U.S. ARMY
RDECOM

TECHNICAL REPORT RDMR-SS-12-05

Least Squares Polynomial Regression with Constraints: A Computer Program

Clark D. Mikkelsen
System Simulation and Development Directorate
Aviation and Missile Research, Development,
and Engineering Center

July 2012

Distribution Statement A: Approved for public release;
distribution is unlimited.

AMRDEC

DESTRUCTION NOTICE

FOR CLASSIFIED DOCUMENTS, FOLLOW THE PROCEDURES IN DoD 5200.22-M, INDUSTRIAL SECURITY MANUAL, SECTION II-19 OR DoD 5200.1-R, INFORMATION SECURITY PROGRAM REGULATION, CHAPTER IX. FOR UNCLASSIFIED, LIMITED DOCUMENTS, DESTROY BY ANY METHOD THAT WILL PREVENT DISCLOSURE OF CONTENTS OR RECONSTRUCTION OF THE DOCUMENT.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL ENDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

REPORT DOCUMENTATION PAGE	**Form Approved** **OMB No. 074-0188**

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY	**2. REPORT DATE** July 2012	**3. REPORT TYPE AND DATES COVERED** Final

4. TITLE AND SUBTITLE Least Squares Polynomial Regression with Constraints: A Computer Program	**5. FUNDING NUMBERS**
6. **AUTHOR**(S) Clark D. Mikkelsen	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Commander, U.S. Army Research, Development, and Engineering Command ATTN: RDMR-SSM-A Redstone Arsenal, AL 35898-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER** TR-RDMR-SS-12-05
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT ***(Maximum 200 Words)***

The least squares polynomial regression constraints program LSPRWC presented in this report is a computer code written to perform a least squares polynomial regression on a given set of data pairs or observations with the option to impose constraints on the regression polynomial. The FORTRAN 77 computer program listing, as well as detailed information on theory, program structure, and limitations, are given along with two sample cases.

14. SUBJECT TERMS least squares, polynomial regression, constraints, curve fit, FORTRAN 77	**15. NUMBER OF PAGES** 110
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	**18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	**19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	**20. LIMITATION OF ABSTRACT** SAR

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

LSPRWC is a FORTRAN 77 computer program written to perform a least squares polynomial regression on a given set of
(x, y) data pairs or observations. In common parlance, LSPRWC, a "curve fit" procedure, finds that n_p order polynomial

$$P(x) = \sum_{j=0}^{n_p} b_j x^j$$

which minimizes the sum of the squared errors, the squared residuals, between the polynomial value $P(x)$ and $y(x)$ over the set of (x, y) data pairs.

In addition, LSPRWC provides for the imposition of constraints on $P(x)$ such that the n_c derivative of $P(x)$ may be specified at a given x location.

LSPRWC was written for LINUX[1]/UNIX[2] users as an analysis tool in science and engineering. The program was intended for users who care about numerical precision, wish to have a well documented source code in hand, and abhor spreadsheet calculations and license fees.

LSPRWC requires command line interaction. No graphical user interfaces are provided with the computer program—although they could certainly be added. The code was not meant for the "point and click" community of computer users.

Least squares polynomial regression is a well known analytical technique dating back to at least the 18th century[3]. The technique remains a premier method for the functional approximation of presumably repeatable phenomena when observations of that phenomena are known to be imperfect. The technique is also commonly used to quantify those phenomena which either defy description by the known laws of physics or for which the mathematical descriptions are simply too unwieldy or numerically intensive for practical application.

Although the functional form for linear regression need not be based on polynomials, polynomials have proven to be particularly useful for a broad range of applications. Additionally, the closed-form mathematical solution for $P(x)$ is readily derived, as given in Appendix A. Once the b_j coefficients are known, polynomial evaluation reduces to a simple numerical process. Furthermore, it is relatively easy to compute derivatives and other properties, such as arc length and integrals for polynomials.

LSPRWC was specifically written for source code control, source code that is readily tailored for specific applications, and to maintain ease of use with the advent of new computational platforms and operating systems. As such, the code eliminates those inexplicable fee requirements for licensed software which exploits an analytical technique embodied in the public domain for centuries.

The addition of constraints was added to LSPRWC as a particularly useful capability not usually included with commercial least squares regression software packages. Often as not, some physical constraint—an initial condition such as zero velocity and a second derivative such as gravitational acceleration—are known absolutely and can be included with LSPWRC to better approximate a given phenomena.

LSPRWC, as written for LINUX/UNIX operating systems, requires only a FORTRAN 77[4] compiler and a terminal window system such as the *X Window System*[5]. The code is known to compile successfully with the Free Software Foundation G77[6] compiler and to work well on *Apple Mac OS X*[7] and *Yellow Dog*[8] LINUX computer operating systems. Although written to FORTRAN 77 standards, the code may compile with FORTRAN 95[9] compilers and does so with the Intel ifort[10]compiler. LSPRWC memory and disk storage requirements are flexible and readily adjusted.

The LSPRWC computer code is not guaranteed to run successfully or operate without failure on any given computer and/or operating system. Considerable error messaging is included in the FORTRAN 77 code, but this does not assure that the linear regression process operates as intended. Hence, it is advised that the user first execute test cases with known solutions for comparison with the LSPRWC code output.

II. FEATURES

The LSPRWC code was written to provide a user friendly environment while offering extensive analytical capabilities by means of the following features:

- Full directives are written to the terminal window whenever input to the LSPRWC code is requested.

- All operational parameters are input using either NAMELIST or free field reads.

- Input to the LSPRWC code from formatted disk files may be offered as an option. In such cases, the data file format, including free field reads, will be input from the command line.

- Sets of (x, y) data pairs are entered either by NAMELIST or read from a formatted disk file.

- Optional constraints are input either by NAMELIST or read from a formatted disk file. Each constraint is defined by the derivative order $n_c = 0$ through n_p - 1, the x location for the constraint, and the value of the n_c derivative of $P(x)$ at the constraint location.

- Computational results from the linear least squares regression are output to the terminal window as the LSPRWC code executes. This output includes the values of the b_j coefficients of the n_p order polynomial $P(x)$ and the standard deviation of the (x, y) data pairs with respect to the regression polynomial.

- As an option, x values may be prescribed for evaluation of the resultant least squares polynomial. These x values may be entered by NAMELIST, input from a formatted disk file, or alternatively as minimum, maximum, and interval values to produce an array of evaluated (x, $P(x)$) data pairs.

- A plot option is included in the LSPRWC code such that the user may visually compare the evaluated (x, $P(x)$) data with the (x, y) data pairs. No plot package is actually included with the LSPRWC code, but the two data sets, (x, $P(x)$) and (x,y), are written to formatted data files for ready access to the user's favorite plot package. The plot option subroutine QPS can also easily be modified by the user to include an in-line plot routine for execution with the LSPRWC code.

- Output of results from a LSPRWC run may optionally be written to a formatted disk file as either the entire set of run parameters, input data, and results or as simply the evaluated data. In the latter case, the evaluated data includes the choices of x, the 0 through n_p - 1 derivative value of $P(x)$, the radius of curvature, and the arc length as columnated data sets.

- Restart options are included in the LSPRWC code allowing for easy modification of operational parameters, for example polynomial order n_p, without having to re-enter all of the input information.

- Constraints are satisfied using the method of undetermined Lagrange Multipliers[11].

- The LSPRWC code includes a matrix inversion subroutine GJEMPS which is based on Gauss-Jordan elimination using a maximum pivot strategy[12]. With some minimal effort, the user may substitute their preferred matrix inversion technique should that be desired.

- All of the FORTRAN 77 array sizes are set in PARAMETER statements within the LSPRWC code making it relatively easy to find and change those array size limits should that be required.

- The LSPRWC code maintains 64-bit precision with IMPLICIT REAL*8 statements in each subroutine as required.

III. SPECIAL FEATURES

The following special features were added to the LSPRWC code to enhance the user friendly environment:

- If a simple Yes/No input is requested, then any upper/lower case variation of Yes, No, Y, or N provides a permissible entry as controlled by the FORTRAN subroutine YNOUS listed in Appendix C.

- Disk files to be used for LSPRWC code input may contain alphanumeric headers such as

```
                 Thermal Data Set

              T(K)                    P(Pa)
  289.26111111111      0.18305580366000E+04
  289.81666666667      0.18967477057200E+04
  290.37222222222      0.19650058020000E+04
  290.92777777778      0.20346428497200E+04
  291.48333333333      0.21070378003200E+04
  292.03888888889      0.21815011780800E+04
  292.59444444444      0.22587224587200E+04
  293.15000000000      0.23373226908000E+04
```

  The disk file data will be read, line by line, and ignored until the prescribed data format is satisfied.

- Command line NAMELIST input can be a nuisance. NAMELIST entries begin in column 2 with either a $ or & sign followed immediately with the NAMELIST name. The entries continue with the name of the variables to be changed from their current values, an equal sign, and the new value to be assigned to the variable. The NAMELIST entry must then end with either a $ or & sign. The FORTRAN subroutines NDRUS and NDWUS, as given in Appendix C, were included in the LSPRWC code to simplify this process.

  If a NAMELIST input is requested, then all variables are written to the terminal window with default values to facilitate cut and paste input using a three-button mouse.

  The starting column for NAMELIST entries is irrelevant.

  NAMELIST entries need not begin with the $ or & sign nor is the NAMELIST name required.

  The NAMELIST entry must end with either the $ or & signs but only if variable values are altered. A simple keyboard return will accomplish a NAMELIST entry with no changes to the variable default values.

- Whenever FORTRAN statements include an option for redirection following an error, e.g.

```
READ(UNIT=4,NML=PARM,ERR=169),
```

  and the source of the error is trivial, for example a misspelled NAMELIST variable name, then provision is included in the LSPRWC code for correction of the error without the need to restart.

IV. PROGRAM STRUCTURE

The structure of the LSPRWC FORTRAN 77 computer code as listed in Appendix B is reasonably straight forward. The program routines and their purposes are as follows:

MAIN—Executive control for program LSPRWC

XYDIS—Subroutine for input of the (x, y) data set or observations

CDS—Subroutine for input of the constraints

PODS—Subroutine for input of the polynomial order

LSPRWC—Subroutine to perform the least squares polynomial regression with constraints

XDPEDS—Subroutine to provide an x-coordinate set for evaluation of the least squares regression polynomial

PES—Polynomial evaluation subroutine

QPS—Quick plot subroutine

FDFOS—Formatted disk file output subroutine

GJEMPS—Matrix inversion subroutine using Gauss-Jordan elimination with maximum pivot strategy

POWER—Subroutine to calculate integer powers of real numbers

FACTRL—Factorial evaluation subroutine

As can be seen, the actual least squares polynomial regression algorithm resides in subroutine LSPRWC. Subroutines CDS, PODS, XDPEDS, and XYDIS provide or define input information for program LSPRWC. Subroutine PES provides output data for program LSPRWC. Subroutines FDFOS and QPS output information from program LSPRWC. Subroutines FACTRL, GJEMPS, and POWER are computational utility routines. A flow path for program LSPRWC is given in Figure 1.

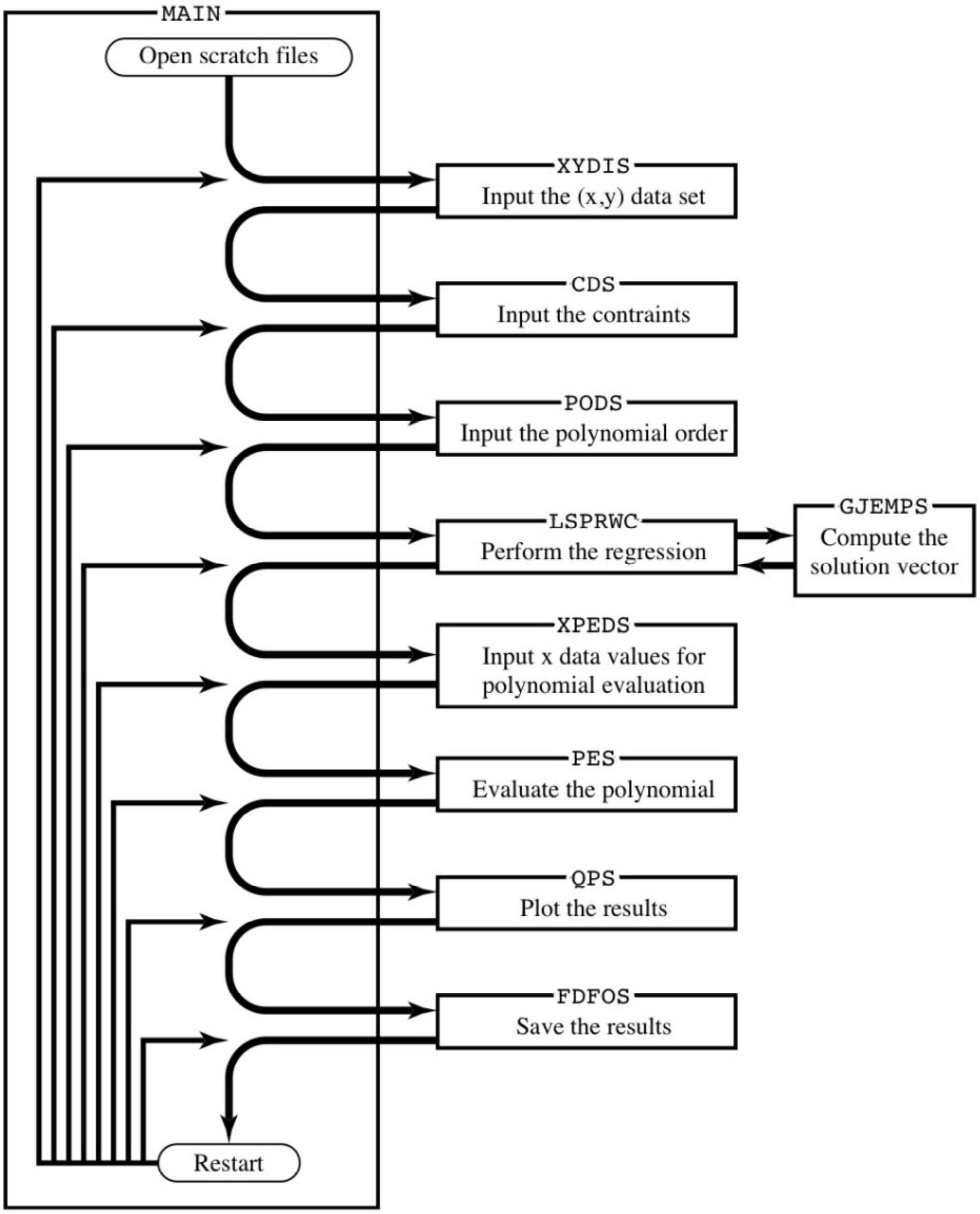


Figure 1. Flow Path for Program LSPRWC

Additional low-level utility routines are listed in the FORTRAN Tool Library in Appendix C.

V. LIMITATIONS

The LSPRWC code has the following requirements or limitations:

- The LSPRWC code was written with the following set of COMMON and PARAMETER statements which set the array sizes consistently in all subroutines.

MNXYDP = Maximum Number of (*x, y*) Data Pairs

= 101

MNXCE = Maximum Number of *x*-Coordinates for polynomial evaluation

= 101

MXANC = Maximum Number of Allowable Constraints

= 20

MXNR = Maximum Number of Rows

= 30

MXNRM1 = Maximum Number of Rows minus 1

= 29

MXNC = Maximum Number of Columns

= 31

MXNCM1 = Maximum Number of Columns minus 1

= 30

The number of rows and columns refers to the size of the augmented matrix for the solution of *n* linear equations in *n* unknowns. It follows from the derivation of Appendix A that

$$\text{MXNR} \geq n_p + \xi + 1, \tag{1}$$

$$\text{MXNRM1} = \text{MXNR} - 1, \tag{2}$$

$$\text{MXNC} = \text{MXNR} + 1, \text{ and} \tag{3}$$

$$\text{MXNCM1} = \text{MXNC} - 1. \tag{4}$$

The PARAMETER variables MXANC and MXNR apply only for NAMELIST input. Any number of (*x, y*) data pairs or *x*-coordinates for polynomial evaluation can be read from disk files.

- The set of (*x, y*) data pairs, as read into program LSPRWC, must be monotonically increasing in *x*. Provision could have been made within program LSPRWC to reorder any set of (*x, y*) data pairs; however, it has been discovered that the failure of a data set to satisfy this requirement is often the indication of unknown errors. Hence, no data set reordering has been implemented.

- The number of (*x*, *y*) data pairs or observations, *m*, must satisfy

$$m \geq n_p + \xi + 1 \; . \tag{5}$$

- The polynomial order n_p must satisfy

  $$0 \leq n_p \leq 9 \; . \tag{6}$$

  The LSPRWC code could, though not readily, be modified for polynomial orders greater than 9; however, this largely defeats the purpose intended for polynomial regression.

- The constraint order n_c must satisfy

  $$0 \leq n_c < n_p \; . \tag{7}$$

- Polynomials $P(x)$ will not produce an infinite slope. Hence, a transformation—possibly a rotation—may be required for a given set of (x, y) data pairs before a least squares polynomial regression can be completed to eliminate that possibility.

VI. ERROR MESSAGES

The LSPRWC code was written with the following extensive embedded error messaging:

- The error messages cover the standard FORTRAN errors and those associated directly with the operation of program LSPRWC.

- Wherever possible, the LSPRWC code identifies the source of the particular error and the subroutine call structure leading to that error.

- Should readily correctable errors occur, for example a misspelled NAMELIST variable name entry, then program LSPRWC will identify the error and loop, in this case for the NAMELIST entry.

- The standard FORTRAN errors source from the FORTRAN OPEN, CLOSE, READ, and WRITE commands.

- Program LSPRWC will issue an error message should the array limits be exceeded. (See Section V.)

- Program LSPRWC will issue an error message should an illegal value for n_p be entered. (See Section V.)

- Program LSPRWC will issue an error message should an illegal value for n_c be entered. (See Section V.)

- Program LSPRWC will issue an error message should the number of (x, y) data pairs be insufficient for the number of constraints, ξ, and the polynomial order, n_p. (See Section V.)

REFERENCES

1. Bovet, D. and Cesati, M., Understanding the LINUX Kernel, 3rd Edition, ISBN-10: 0-596-00565-2, O'Reilly Media, Inc., 2005.

2. Rosen, K. et al., UNIX: The Complete Reference, 2nd Edition, ISBN-10: 0-07-226336-9, MCGRAW-HILL, 2007.

3. Wikipedia, "Least squares," http://en.wikipedia.org/wiki/Least_squares/

4. FORTRAN 77 Language Reference Manual, Document Number 007-0710-030, Silicon Graphics, Inc., 1990.

5. Young, D.A., The X Window System: Programming and Applications with Xt, OSF/Motif, 2nd Edition, ISBN 0-13-123803-5, Prentice-Hall, 1994.

6. http://gcc.gnu.org/fortran/

7. Wikipedia, "Mac OS X," http://en.wikipedia.org/wiki/Mac_OS_X

8. Wikipedia, "Yellow Dog Linux," http://en.wikipedia.org/wiki/Yellow_Dog_Linux

9. Adams, J.C. et al., FORTRAN 95 HANDBOOK, ISBN 0-262-51096-0, The MIT Press, 1997.

10. "Intel Fortran Compiler for Linux," http://software.intel.com/en-us/articles/intel-fortrancompiler-for-linux-9x-manuals/

11. Weinstock, R., Calculus of Variations with Applications to Physics and Engineering, New York, Dover Publications, Inc., pp. 6, 1974.

12. Carnahan, B.; Luther, H.A.; and Wilkes, J.L., Applied Numerical Methods, New York, John Wiley and Sons, pp. 281-96, 1969.

APPENDIX A
DERIVATION

A set of (x,y) data pairs are to be "curve fit" using the method of least squares; that is, an n_p order polynomial is chosen such that the sum of the squared errors is minimized over the set of (x,y) data pairs. In addition, the n_p order polynomial must satisfy a set of constraints; namely, that the n_c polynomial derivative be specified at a given *x* location.

Then with the n_p order polynomial given by

$$P(x) = \sum_{j=0}^{n_p} b_j x^j, \tag{1}$$

the squared error to be minimized is

$$S = \sum_{i=1}^{m} [y_i - P(x_i)]^2, \tag{2}$$

$$= \sum_{i=1}^{m} (y_i - \sum_{j=0}^{n_p} b_j x_i^j)^2, \tag{3}$$

subject to the constraints

$$G_l(x_l) = P^{k_l}(x_l) = C_l \tag{4}$$

$$= \sum_{j=k_l}^{n_p} \frac{j!}{(j-k_l)!} b_j x_l^{j-k} = C_l, \tag{5}$$

$$\text{for } \; l = 1, 2, 3, \cdots, \xi$$

where

n_p = polynomial order,
ξ = number of constraints,
m = number of (x,y) data pairs,
x_l = l_{th} constraint location,
k_l = l_{th} constraint order $(0 \le k_l \le n_p)$,
C_l = value of the l_{th} constraint.

Using the method of undetermined Lagrange multipliers, the relation to be minimized is

$$S^* = S + \sum_{l=1}^{\xi} \lambda_l G_l \tag{6}$$

by setting

$$\frac{\partial S^*}{\partial b_j} = 0 \quad \text{for} = 0, 1, \cdots, n_p. \tag{7}$$

From equations (3), (5), and (6),

$$S^* = \sum_{i=1}^{m} (y_i - \sum_{j=0}^{n_p} b_j x_i^j)^2 + \sum_{l=1}^{\xi} \lambda_l \sum_{j=k_l}^{n_p} \frac{j!}{(j-k_l)!} b_j x_l^{j-k_l}. \tag{8}$$

Then using equation (7),

$$\frac{\partial S^*}{\partial b_J} = \sum_{i=1}^{m} (-2x_i^J)(y_i - \sum_{j=0}^{n_p} b_j x_i^j) + \sum_{l=1}^{\xi} \lambda_l F_{l,J} = 0 \tag{9}$$

where

$$F_{l,J} = \begin{cases} \frac{J!}{(J-k_l)!} x_l^{J-k_l} & , J \geq k_l \\ 0 & , J < k_l \end{cases} . \tag{10}$$

Rewriting equation (9) gives

$$\sum_{i=1}^{m} \sum_{j=0}^{n_p} b_j x_i^{J=j} + \frac{1}{2} \sum_{l=1}^{\xi} \lambda_l F_{l,J} = \sum_{i=1}^{m} y_i x_i^J \tag{11}$$

$$\text{for } J = 0, 1, \cdots, n_p .$$

The minimization equation (11) provides $n_p + 1$ equations in the $n_p + 1$ unknown coefficients b_j and ξ unknown Lagrange multipliers λ_l. Constraint equation (5) provides the remaining ξ equations in the unknown b_j. Since equations (5) and (11) are linear and algebraic, they may be solved simultaneously using a matrix inversion technique.

APPENDIX B
FORTRAN 77 CODE

```
C***********************************************************************
C*                                                                     *
C*    LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS (LSPRWC)    *
C*                                                                     *
C*                      WRITTEN BY: C.D. MIKKELSEN                     *
C*                                                                     *
C*                             2 MAY 1988                              *
C*                                                                     *
C*           AERODYNAMICS TECHNOLOGY BRANCH (AMSMI-RD-SS-AT)           *
C*            SYSTEMS SIMULATION AND DEVELOPMENT DIRECTORATE           *
C*    US ARMY MISSILE RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER    *
C*                       US ARMY MISSILE COMMAND                       *
C*                REDSTONE ARSENAL, ALABAMA 35898-5252                 *
C*                                                                     *
C* REF: CARNAHAN,B., LUTHER, H.A., AND WILKES, J.O.: APPLIED NUMERICAL *
C*      METHODS, NEW YORK, JOHN WILEY & SONS, 1969, PP. 571-584.       *
C*                                                                     *
C*      WEINSTOCK, R.: CALCULUS OF VARIATIONS WITH APPLICATIONS TO     *
C*      PHYSICS AND ENGINEERING, NEW YORK, DOVER PUBLICATIONS, INC.,   *
C*      1974, P. 6.                                                    *
C*                                                                     *
C*                     REVISION DATE: 7 JULY 2011                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*         PROGRAM LSPRWC REQUIRES THE FOLLOWING SUBPROGRAMS:          *
C*                                                                     *
C*             CBUS06 CDS DFCUS FACTRL FDFOS GJEMPS LSPRWC             *
C*             PES  PODS  POWER  QPS  XDPEDS  XYDIS  YNOUS             *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)         *
C*                                                                     *
C* LOGICAL UNIT DEFINITIONS:                                           *
C*                                                                     *
C* UNIT FILE                                                           *
C*                                                                     *
C*   1  INPUT (X,Y) DATA PAIR SCRATCH FILE                             *
C*   2  (XC,NC,CV) CONSTRAINT SCRATCH FILE                             *
C*   3  EVALUATED DATA SCRATCH FILE                                    *
C*   4  TEMPORARY USE                                                  *
C*   5  STANDARD INPUT                                                 *
C*   6  STANDARD OUTPUT                                                *
C*                                                                     *
C***********************************************************************
C
C
      PROGRAM MAIN
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER CCV*1
C
```

```
      INTEGER DUM2
C
      PARAMETER (MXNRM1=29)
C
      COMMON/B/B(0:MXNRM1)
      COMMON/NCONST/NCONST
      COMMON/NEVAL/NEVAL
      COMMON/NP/NP
      COMMON/SDEV/SDEV
C
C
C***********************************************************************
C*                                                                     *
C*                       OPEN THE SCRATCH FILES                        *
C*                                                                     *
C***********************************************************************
C
C
      OPEN(UNIT=1,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=116)
      OPEN(UNIT=2,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=117)
      OPEN(UNIT=3,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=118)
C
C
C***********************************************************************
C*                                                                     *
C*                    WRITE THE PROGRAM DESCRIPTION                    *
C*                                                                     *
C***********************************************************************
C
C
      WRITE(UNIT=6,FMT=201)
      WRITE(UNIT=6,FMT=202)
      READ(UNIT=5,FMT=203)CCV
C
C
C***********************************************************************
C*                                                                     *
C*                INPUT THE X-Y DATA POINTS TO BE FITTED               *
C*                                                                     *
C***********************************************************************
C
C
101   CALL XYDIS(*119,*115)
C
C
C***********************************************************************
C*                                                                     *
C*                        INPUT THE CONSTRAINTS                        *
C*                                                                     *
C***********************************************************************
C
C
102   NCONST=0
103   WRITE(UNIT=6,FMT=208)
      CALL YNOUS(*104,*105,*103)
104   CALL CDS(*120,*115)
C
C
C***********************************************************************
C*                                                                     *
```

```
C*                     INPUT THE POLYNOMIAL ORDER                      *00000121
C*                                                                     *00000122
C***********************************************************************00000123
C                                                                       00000124
C                                                                       00000125
105   CALL PODS(*115)                                                   00000126
C                                                                       00000127
C                                                                       00000128
C***********************************************************************00000129
C*                                                                     *00000130
C*  PERFORM THE LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS   *00000131
C*                                                                     *00000132
C***********************************************************************00000133
C                                                                       00000134
C                                                                       00000135
      CALL LSPRWC(*121)                                                 00000136
      WRITE(UNIT=6,FMT=209)                                             00000137
      WRITE(UNIT=6,FMT=210)(I,B(I),I=0,NP)                              00000138
      WRITE(UNIT=6,FMT=211)NP,SDEV                                      00000139
      WRITE(UNIT=6,FMT=202)                                             00000140
      READ(UNIT=5,FMT=203)CCV                                           00000141
C                                                                       00000142
C                                                                       00000143
C***********************************************************************00000144
C*                                                                     *00000145
C*                INPUT THE X DATA POINTS FOR EVALUATION               *00000146
C*                   OF THE LEAST SQUARES POLYNOMIAL                   *00000147
C*                                                                     *00000148
C***********************************************************************00000149
C                                                                       00000150
C                                                                       00000151
106   WRITE(UNIT=6,FMT=212)                                             00000152
      LCV2=0                                                            00000153
      CALL YNOUS(*107,*112,*106)                                        00000154
107   CALL XDPEDS(LCV2,*122,*115)                                       00000155
C                                                                       00000156
C                                                                       00000157
C***********************************************************************00000158
C*                                                                     *00000159
C*  EVALUATE THE LEAST SQUARES POLYNOMIAL AT THE PRESCRIBED X VALUES   *00000160
C*                                                                     *00000161
C***********************************************************************00000162
C                                                                       00000163
C                                                                       00000164
108   CALL PES(*123)                                                    00000165
109   WRITE(UNIT=6,FMT=205)                                             00000166
      CALL YNOUS(*110,*112,*109)                                        00000167
110   REWIND 3                                                          00000168
      WRITE(UNIT=6,FMT=206)                                             00000169
      DO 111 I=1,NEVAL                                                  00000170
      READ(UNIT=3,ERR=124)X,Y                                           00000171
      IF(MOD(I,20).NE.0) GO TO 111                                      00000172
      WRITE(UNIT=6,FMT=202)                                             00000173
      READ(UNIT=5,FMT=203)CCV                                           00000174
      WRITE(UNIT=6,FMT=206)                                             00000175
111   WRITE(UNIT=6,FMT=207)I,X,Y                                        00000176
      WRITE(UNIT=6,FMT=202)                                             00000177
      READ(UNIT=5,FMT=203)CCV                                           00000178
C                                                                       00000179
C                                                                       00000180
```

```
C***********************************************************************
C*                                                                     *
C*                          PLOT THE RESULTS                           *
C*                                                                     *
C***********************************************************************
C
C
112   WRITE(UNIT=6,FMT=213)
      CALL YNOUS(*113,*115,*112)
113   CALL QPS(*125)
      GO TO 115
C
C
C***********************************************************************
C*                                                                     *
C*                 SAVE RESULTS ON FORMATTED DISK FILE                 *
C*                                                                     *
C***********************************************************************
C
C
114   CALL FDFOS(LCV2,*126)
C
C
C***********************************************************************
C*                                                                     *
C*                          RESTART PROCEDURE                          *
C*                                                                     *
C***********************************************************************
C
C
115   WRITE(UNIT=6,FMT=214)
      READ(UNIT=5,FMT=204)LCV1
      GO TO (101,104,105,107,113,114),LCV1
      STOP
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
116   WRITE(UNIT=6,FMT=215)
      STOP
117   WRITE(UNIT=6,FMT=216)
      STOP
118   WRITE(UNIT=6,FMT=217)
      STOP
119   WRITE(UNIT=6,FMT=218)
      STOP
120   WRITE(UNIT=6,FMT=219)
      STOP
121   WRITE(UNIT=6,FMT=220)
      GO TO 128
122   WRITE(UNIT=6,FMT=221)
      STOP
123   WRITE(UNIT=6,FMT=222)
      STOP
124   WRITE(UNIT=6,FMT=223)
```

```
      STOP
125   WRITE(UNIT=6,FMT=224)
      STOP
126   WRITE(UNIT=6,FMT=225)
      STOP
127   WRITE(UNIT=6,FMT=226)
      STOP
128   WRITE(UNIT=6,FMT=202)
      READ(UNIT=5,FMT=203)CCV
      GO TO 115
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,T4,'PROGRAM LSPRWC IS AN INTERACTIVE FORTRAN PROGRAM TO P
     >ERFORM A LEAST',/,'SQUARES POLYNOMIAL REGRESSION WITH CONTRAINTS; 
     >THAT IS, A SET OF X-Y DATA',/,'POINTS IS CURVE FIT WITH AN NP ORDE
     >R POLYNOMIAL OF THE FORM',//,T17,'P(X)=B0+B1*X+B2*X**2+B3*X**3+...
     >.+BNP*X**NP',//,'WITH ANY POLYNOMIAL DERIVATIVES, ZERO THROUGH NP,
     > SPECIFIED AT GIVEN X',/,'LOCATIONS.  THE PROCEDURE FOLLOWED IS TH
     >E METHOD OF LEAST SQUARES USING',/,'UNDETERMINED LAGRANGE MULTIPLI
     >ERS.',//,T4,'AS AN INTERACTIVE PROGRAM, LSPRWC IS SELF-EXPLANATORY
     > AND PROMPTS FOR THE',/,'NECESSARY INFORMATION.  THE X-Y DATA TO B
     >E FITTED MAY BE ENTERED BY NAMELIST',/,'OR READ FROM A FORMATTED D
     >ISC FILE.  PROGRAM LSPRWC WILL ALSO EVALUATE THE',/,'RESULTANT LEA
     >ST SQUARES POLYNOMIAL AT PRESCRIBED VALUES OF X WHICH, AGAIN,',/,'
     >MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE.')
202   FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')
203   FORMAT(A1)
204   FORMAT(I1)
205   FORMAT(/,'SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
     >')
206   FORMAT(/,T10,'NO.',T26,'X',T49,'Y')
207   FORMAT(T8,I5,1P2D23.13)
208   FORMAT(/,'ARE CONSTRAINTS DESIRED? (Y/N)')
209   FORMAT(T7,'LEAST SQUARES POLYNOMIAL',//,T19,'P(X)=B(0)+B(1)*X+B(2)
     >*X**2+....+B(NP)*X**NP',//,T22,'I',T36,'B(I)',/)
210   FORMAT(' ',T22,I1,1PD25.13)
211   FORMAT(/,T4,'THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER 
     > ',I1,' IS ',1PD12.5)
212   FORMAT(/,'SHOULD DATA POINTS BE PRESCRIBED FOR EVALUATION OF THE L
     >EAST SQUARES',/,'POLYNOMIALS? (Y/N)')
213   FORMAT(/,'SHOULD THE RESULTS OF THIS RUN BE QUICK-PLOTTED? (Y/N)')
     >
214   FORMAT(/,'ENTER:',/,'1, TO RESTART THE PROGRAM',/,'2, TO CHANGE TH
     >E CONSTRAINTS',/,'3, TO CHANGE THE POLYNOMIAL ORDER',/,'4, TO CHAN
     >GE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL',/,'5,
     > TO PLOT THE RESULTS',/,'6, TO SAVE THE RESULTS ON A FORMATTED DIS
     >C FILE',/,'7, TO STOP')
215   FORMAT(/,'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 1, STATUS = 
     >"SCRATCH"')
216   FORMAT(/,'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 2, STATUS = 
     >"SCRATCH"')
217   FORMAT(/,'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 3, STATUS = 
     >"SCRATCH"')
```

```
218   FORMAT('SUBROUTINE XYDIS WAS CALLED FROM PROGRAM LSPRWC')
219   FORMAT('SUBROUTINE CDS WAS CALLED FROM PROGRAM LSPRWC')
220   FORMAT('SUBROUTINE LSPRWC WAS CALLED FROM PROGRAM LSPRWC')
221   FORMAT('SUBROUTINE XDPEDS WAS CALLED FROM PROGRAM LSPRWC')
222   FORMAT('SUBROUTINE PES WAS CALLED FROM PROGRAM LSPRWC')
223   FORMAT(/,'ERROR IN PROGRAM LSPRWC: UNFORMATTED READ ERROR ON$UNIT
     >3')
224   FORMAT('SUBROUTINE QPS WAS CALLED FROM PROGRAM LSPRWC')
225   FORMAT('SUBROUTINE FDFOS WAS CALLED FROM PROGRAM LSPRWC')
226   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM PROGRAM LSPRWC')
      END
      SUBROUTINE CDS(*,*)
C
C
C***********************************************************************
C*                                                                     *
C*               CONSTRAINT DEFINITION SUBROUTINE (CDS)                *
C*                                                                     *
C*                     REVISION DATE: 1 JULY 2011                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*       SUBROUTINE CDS DEFINES CONSTRAINTS FOR PROGRAM LSPRWC.        *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* CV     = VALUE OF THE NC-TH DERIVATIVE FOR THE CONSTRAINT AT XC    *
C* NC     = CONSTRAINT ORDER                                           *
C* NCONST = NUMBER OF CONSTRAINTS                                      *
C* XC     = X-VALUE FOR THE CONSTRAINT                                 *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXNAC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS                    *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER CCV*1,DFmt*3,File*80,Fmt*80
C
      INTEGER DUM2
C
      PARAMETER (R_NaN=-999.E+00)
      PARAMETER (MXNAC=20)
C
      COMMON/CV/CV(MXNAC)
      COMMON/NC/NC(MXNAC)
      COMMON/NCONST/NCONST
      COMMON/XC/XC(MXNAC)
C
      DATA DFmt/'(*)'/
C
      NAMELIST/PARM/CV,NC,XC
C
C
C***********************************************************************
C*                                                                     *
C*                        INPUT THE CONSTRAINTS                        *
C*                                                                     *
C***********************************************************************
C
C
      DO 101 I=1,MXNAC
      CV(I)=R_NaN
      NC(I)=-1
101   XC(I)=R_NaN
      WRITE(UNIT=6,FMT=201)
      READ(UNIT=5,FMT=202)LCV1
      IF(LCV1.EQ.2) GO TO 104
C
102   WRITE(UNIT=6,FMT=203)MXNAC
      WRITE(UNIT=6,FMT=204)
      CALL NDRUS('PARM',4,*102,*121)
      READ(UNIT=4,NML=PARM,ERR=102)
      CALL NDRUSE(*102,*121)
      DO 103 NCONST=MXNAC,1,-1
      IF(XC(NCONST).NE.R_NaN) GO TO 111
103   CONTINUE
      GO TO 111
C
104   WRITE(UNIT=6,FMT=205)MXNAC
      WRITE(UNIT=6,FMT=206)
      READ(UNIT=5,FMT=207)FILE
105   WRITE(UNIT=6,FMT=208)
      WRITE(UNIT=6,FMT=209)
      CALL DFCUS(Fmt,DFmt,*105)
      OPEN(UNIT=4,FILE=File,STATUS='OLD',ERR=122)
      REWIND 4
      IF(Fmt(1:3).EQ.DFmt) GO TO 107
      DO 106 NCONST=1,MXNAC
106   READ(UNIT=4,FMT=Fmt,END=109,ERR=123)XC(NCONST),NC(NCONST),
     >CV(NCONST)
      GO TO 110
107   DO 108 NCONST=1,MXNAC
108   READ(UNIT=4,FMT=*,END=109,ERR=123)XC(NCONST),NC(NCONST),CV(NCONST)
      GO TO 110
109   NCONST=NCONST-1
110   CLOSE(UNIT=4,STATUS='KEEP',ERR=124)
C
C
C***********************************************************************
C*                                                                     *
C*                    SORT AND EDIT THE CONSTRAINTS                    *
C*                                                                     *
C***********************************************************************
C
C
111   IF(NCONST.EQ.0) GO TO 120
      IF(NCONST.EQ.1) GO TO 117
      DO 112 I=1,NCONST-1
      DO 112 J=I+1,NCONST
      IF(XC(I).NE.XC(J)) GO TO 112
      IF(NC(I).NE.NC(J)) GO TO 112
      XC(I)=R_NaN
112   CONTINUE
113   LCV1=0
      DO 115 I=2,NCONST
      IF(XC(I-1).LT.XC(I)) GO TO 115
      IF(XC(I-1).GT.XC(I)) GO TO 114
      IF(NC(I-1).LT.NC(I)) GO TO 115
114   LCV1=1
      DUM1=XC(I-1)
      DUM2=NC(I-1)
      DUM3=CV(I-1)
      XC(I-1)=XC(I)
      NC(I-1)=NC(I)
      CV(I-1)=CV(I)
      XC(I)=DUM1
      NC(I)=DUM2
      CV(I)=DUM3
115   CONTINUE
      IF(LCV1.EQ.1) GO TO 113
      DO 116 NCONST=MXNAC,1,-1
      IF(XC(NCONST).NE.R_NaN) GO TO 117
116   CONTINUE
C
117   WRITE(UNIT=6,FMT=210)
      CALL YNOUS(*118,*120,*117)
118   WRITE(UNIT=6,FMT=211)
      DO 119 I=1,NCONST
      IF(MOD(I,20).NE.0) GO TO 119
      WRITE(UNIT=6,FMT=212)
      READ(UNIT=5,FMT=213)CCV
      WRITE(UNIT=6,FMT=211)
119   WRITE(UNIT=6,FMT=214)I,XC(I),NC(I),CV(I)
      WRITE(UNIT=6,FMT=212)
      READ(UNIT=5,FMT=213)CCV
120   RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
121   WRITE(UNIT=6,FMT=215)
      GO TO 126
122   WRITE(UNIT=6,FMT=216)
      CALL CBUS06(6,File,*125)
      RETURN 1
123   WRITE(UNIT=6,FMT=217)
      CALL CBUS06(6,File,*125)
      RETURN 1
124   WRITE(UNIT=6,FMT=218)
      CALL CBUS06(6,File,*125)
      RETURN 1
125   WRITE(UNIT=6,FMT=219)
      RETURN 1
126   WRITE(UNIT=6,FMT=212)
      READ(UNIT=5,FMT=213)CCV
      RETURN 2
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE CONSTRAINTS FROM THE
     >FOLLOWING LIST:',//,'1, FOR KEYBOARD INPUT VIA NAMELIST',/,'2, FOR
     > FORMATTED DISK FILE INPUT',//,'NOTE: THE CONSTRAINTS ARE COMPLETE
     >LY INDEPENDENT OF THE X-Y DATA POINTS TO BE',/,'FITTED.')
202   FORMAT(I1)
```

```
203   FORMAT(/,'INPUT THE CONSTRAINTS (',I2,' MAX) BY NAMELIST WHERE:',/
     >/,'CV     = ARRAY OF POLYNOMIAL DERIVATIVE VALUES',/,'NC     = ARR
     >AY OF POLYNOMIAL DERIVATIVE ORDERS (CONSTRAINT ORDER)',/,'XC     =
     > ARRAY OF POLYNOMIAL DERIVATIVE LOCATIONS (CONSTRAINT X VALUES)',/
     >/,'NOTE: NC VALUES MUST BE IN THE RANGE 0 TO NP.',//,T2,'CURRENT V
     >ALUES ARE:')
204   FORMAT(/,'$PARM CV=____,____,____,NC=____,____,____,XC=____,____,_
     >___$',/)
205   FORMAT(/,'CONSTRAINTS (',I2,' MAX) ARE INPUT FROM FORMATTED DISK F
     >ILES AS (XC,NC,CV) TRIPLES',/,'WHERE:',//,'CV     = POLYNOMIAL DER
     >IVATIVE VALUE',/,'NC     = POLYNOMIAL DERIVATIVE OR CONSTRAINT ORD
     >ER',/,'XC     = POLYNOMIAL DERIVATIVE LOCATION (CONSTRAINT X VALUE
     >)',//,'NOTE: NC VALUES MUST BE IN THE RANGE 0 TO NP.')
206   FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
     >')
207   FORMAT(A80)
208   FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)')
209   FORMAT('NOTE: AN EXAMPLE FORMAT IS "(2E15.6)"',/,T7,'ENTER"(*)" FO
     >R A FREE FIELD READ (DEFAULT FORMAT)')
210   FORMAT(/,'SHOULD THE CONSTRAINTS BE DISPLAYED FOR VERIFICATION? (Y
     >/N)')
211   FORMAT(/,T6,'NO.',T27,'XC(I)',T35,'NC(I)',T58,'CV(I)')
212   FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')
213   FORMAT(A1)
214   FORMAT(T8,I3,1PD23.13,I7,D23.13)
215   FORMAT('SUBROUTINE NDRUS WAS CALLED FROM PROGRAM LSPRWC')
216   FORMAT(/,'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 4, STATUS =
     >"OLD", FILE =',/)
217   FORMAT(/,'ERROR IN PROGRAM LSPRWC: READ ERROR ON UNIT 4, FILE =',/
     >)
218   FORMAT(/,'ERROR IN PROGRAM LSPRWC: CLOSE ERROR ON UNIT 4, FILE =',
     >/)
219   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM PROGRAM LSPRWC')
      END
      FUNCTION FACTRL(N)
C
C
C***********************************************************************
C*                                                                     *
C*          FACTORIAL EVALUATION FUNCTION SUBPROGRAM (FACTRL)          *
C*                                                                     *
C*                     REVISION DATE: 15 JUNE 1989                     *
C*                                                                     *
C***********************************************************************
C
C
      INTEGER FACTRL
      FACTRL=1
      IF(N.LE.1) RETURN
      DO 101 I=0,N-1
101   FACTRL=FACTRL*(N-I)
      END
      SUBROUTINE FDFOS(LCV2,*)
C
C
C***********************************************************************
C*                                                                     *
C*            FORMATTED DISK FILE OUTPUT SUBROUTINE (FDFOS)            *
C*                                                                     *
C*                    REVISION DATE: 2 OCTOBER 2007                    *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXANC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS                    *
C* MXNR   = MAXIMUM NUMBER OF ROWS                                     *
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)         *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER Alist*1,Blank*1,DFmt*9,File*80,Fmt*80,SList*1
C
      PARAMETER (MXANC=20,MXNRM1=29)
C
      COMMON/B/B(0:MXNRM1)
      COMMON/CV/CV(MXANC)
      COMMON/NC/NC(MXANC)
      COMMON/NCONST/NCONST
      COMMON/NDATA/NDATA
      COMMON/NEVAL/NEVAL
      COMMON/NP/NP
      COMMON/SDEV/SDEV
      COMMON/XC/XC(MXANC)
C
      DIMENSION AList(13),NList(13),SList(-1:11),YK(-1:11)
C
      SAVE Blank,DFmt
C
      DATA Blank/' '/,DFmt/'(13E15.6)'/,SList/"X","0","1","2","3","4",
     >"5","6","7","8","9","R","S"/
C
      NAMELIST/PARM/AList
C
C
C***********************************************************************
C*                                                                     *
C*                        SELECT AN OUTPUT OPTION                      *
C*                                                                     *
C***********************************************************************
C
C
      WRITE(UNIT=6,FMT=201)
      READ(UNIT=5,FMT=202)FILE
      WRITE(UNIT=6,FMT=203)
      READ(UNIT=5,FMT=204)LCV1
      IF(LCV1.EQ.2.AND.LCV2.EQ.0) GO TO 113
      OPEN(UNIT=4,FILE=File,FORM='FORMATTED',STATUS='UNKNOWN',ERR=114)
      REWIND 4
      IF(LCV1.EQ.2) GO TO 104
C
C
C***********************************************************************
C*                                                                     *
C*                  OUTPUT ALL INPUT DATA AND RESULTS                  *
C*                                                                     *
C***********************************************************************
C
C
      WRITE(UNIT=4,FMT=205,ERR=115)
      WRITE(UNIT=4,FMT=206,ERR=115)
      REWIND 1
      DO 101 I=1,NDATA
      READ(UNIT=1,ERR=116)X,Y
101   WRITE(UNIT=4,FMT=207,ERR=115)I,X,Y
      IF(NCONST.EQ.0) GO TO 102
      WRITE(UNIT=4,FMT=208,ERR=115)
      WRITE(UNIT=4,FMT=209,ERR=115)(I,XC(I),NC(I),CV(I),I=1,NCONST)
102   WRITE(UNIT=4,FMT=210,ERR=115)
      WRITE(UNIT=4,FMT=211,ERR=115)(I,B(I),I=0,NP)
      WRITE(UNIT=4,FMT=212,ERR=115)NP,SDEV
      IF(LCV2.EQ.0) GO TO 112
      WRITE(UNIT=4,FMT=213,ERR=115)
      REWIND 3
      DO 103 I=1,NEVAL
      READ(UNIT=3,ERR=117)X,Y
103   WRITE(UNIT=4,FMT=207,ERR=115)I,X,Y
      GO TO 112
C
C
C***********************************************************************
C*                                                                     *
C*                   OUTPUT ONLY THE EVALUATED DATA                    *
C*                                                                     *
C***********************************************************************
C
C
104   DO 105 K=3,13
      AList(K)=Blank
105   NList(K)=-999
      AList(1)='X'
      AList(2)='0'
      WRITE(UNIT=6,FMT=214)SList(-1)
      WRITE(UNIT=6,FMT=215)(SList(K),K,K=0,NP)
      WRITE(UNIT=6,FMT=216)(SList(K),K=10,11)
      WRITE(UNIT=6,FMT=217)
      CALL NDRUS('PARM',9,*104,*118)
      READ(UNIT=9,NML=PARM,ERR=104)
      CALL NDRUSE(*104,*118)
      DO 106 NL=1,13
      IF(AList(NL).EQ.Blank) GO TO 107
106   CONTINUE
107   NL=NL-1
      IF(NL.EQ.0) GO TO 119
      DO 110 I=1,NL
      DO 108 K=-1,11
      IF(AList(I).EQ.SList(K)) GO TO 109
```

```
108   CONTINUE
      GO TO 120
109   NList(I)=K
      IF(K.EQ.10) NList(I)=NP+1
      IF(K.EQ.11) NList(I)=NP+2
      IF(NP.LT.K.AND.K.LT.10) GO TO 120
110   CONTINUE
C
      WRITE(UNIT=6,FMT=218)
      CALL CBUS06(6,DFmt,*122)
      CALL DFCUS(Fmt,DFmt,*104)
      REWIND 3
      DO 111 I=1,NEVAL
      READ(UNIT=3,ERR=117)(YK(K),K=-1,NP+2)
111   WRITE(UNIT=4,FMT=Fmt,ERR=115)(YK(NList(K)),K=1,NL)
112   CLOSE(UNIT=4,STATUS='KEEP',ERR=121)
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
113   WRITE(UNIT=6,FMT=219)
      CALL CBUS06(6,File,*122)
      WRITE(UNIT=6,FMT=220)
      RETURN
114   WRITE(UNIT=6,FMT=221)
      CALL CBUS06(6,File,*122)
      RETURN 1
115   WRITE(UNIT=6,FMT=222)
      CALL CBUS06(6,File,*122)
      RETURN 1
116   WRITE(UNIT=6,FMT=223)
      RETURN 1
117   WRITE(UNIT=6,FMT=224)
      RETURN 1
118   WRITE(UNIT=6,FMT=225)
      RETURN 1
119   WRITE(UNIT=6,FMT=226)
      GO TO 104
120   WRITE(UNIT=6,FMT=227)I,AList(I)
      GO TO 104
121   WRITE(UNIT=6,FMT=228)
      CALL CBUS06(6,File,*122)
      RETURN 1
122   WRITE(UNIT=6,FMT=229)
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
```

```
201   FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE.')
202   FORMAT(A80)
203   FORMAT(/,'SELECT THE TYPE OF OUTPUT FOR THE FORMATTED DISK FILE FR
     >OM THE FOLLOWING LIST:',/,'1, FOR ALL INPUT DATA AND RESULTS',/,'2
     >, FOR ONLY THE EVALUATED DATA')
204   FORMAT(I1)
205   FORMAT(T6,'LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS (L
     >SPRWC)',//,T22,'WRITTEN BY: C.D. MIKKELSEN',//,T31,'2 MAY 1988',//
     >,T13,'AERODYNAMICS TECHNOLOGY BRANCH (AMSMI-RD-SS-AT)',/,T13,'SYST
     >EMS SIMULATION AND DEVELOPMENT DIRECTORATE',/,T6,'US ARMY MISSILE
     >RESEARCH, ENGINEERING, AND DEVELOPMENT CENTER',/,T25,'US ARMY MISS
     >ILE COMMAND',/,T18,'REDSTONE ARSENAL, ALABAMA 35898-5252')
206   FORMAT(/,T20,'X-Y DATA POINTS TO BE FITTED:',//,T12,'NO.',T28,'X',
     >T51,'Y')
207   FORMAT(T10,I5,1P2D23.13)
208   FORMAT(/,T25,'POLYNOMIAL CONSTRAINTS',//,T8,'NO.',T29,'XC(I)',T37,
     >'NC(I)',T60,'CV(I)')
209   FORMAT(T8,I3,1PD23.13,I7,D23.13)
210   FORMAT(/,T28,'LEAST SQUARES POLYNOMIAL',//,T15,'P(X)=B(0)+B(1)*X+B
     >(2)*X**2+....+B(NP)*X**NP',//,T23,'I',T37,'B(I)',/)
211   FORMAT(T23,I1,1PD25.13)
212   FORMAT(/,T1,'THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER '
     >,I1,' IS ',1PD12.5)
213   FORMAT(/,T22,'X-Y EVALUATED DATA POINTS:'//,T12,'NO.',T28,'X',T51,
     >'Y')
214   FORMAT(/,'INPUT THE EVALUATED DATA VARIABLE LIST FOR OUTPUT BY NAM
     >ELIST WHERE:',//,'"',A1,'"    = X VALUES')
215   FORMAT('"',A1,'"    = ',I1,'th DERIVATIVE OF Y AT X')
216   FORMAT('"',A1,'"    = RADIUS OF CURVATURE AT X',/,'"',A1,'"    = A
     >RC LENGTH',//,'NOTE: VARIABLES WILL BE LISTED IN THE ORDER OF INPU
     >T',//,'CURRENT VALUES ARE:')
217   FORMAT(/,'$PARM AList= "X","0"$',/)
218   FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)',//,'NO
     >TE: AN EXAMPLE FORMAT IS "(F15.7,E15.6)"',/,T7,'THE DEFAULT FORMAT
     > IS:')
219   FORMAT(/,'WARNING IN SUBROUTINE FDFOS: NO DATA POINTS HAVE BEEN PR
     >ESCRIBED FOR EVALUATION',/,'OF THE LEAST SQUARES POLYNOMIAL. HENCE
     >, FILE',/)
220   FORMAT(/,'WILL NOT BE OPENED.')
221   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: OPEN ERROR ON UNIT 4, STATUS
     >= "UNKNOWN", FILE =',/)
222   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: WRITE ERROR ON UNIT 4, FILE =
     >',/)
223   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: UNFORMATTED READ ERROR ON UNI
     >T 1')
224   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: UNFORMATTED READ ERROR ON UNI
     >T 3')
225   FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE FDFOS')
226   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: THE OUTPUT LIST MUST NOT BE E
     >MPTY')
227   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: ILLEGAL VALUE FOR',/,'AList('
     >,I2,') = ',A1)
228   FORMAT(/,'ERROR IN SUBROUTINE FDFOS: CLOSE ERROR ON UNIT 4, FILE =
     >',/)
229   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE FDFOS')
      END
      SUBROUTINE GJEMPS(EPS,INDIC,N,DETER,*)
C
C
C***********************************************************************
C*                                                                     *
C*    GAUSS-JORDAN ELIMINATION USING MAXIMUM PIVOT STRATEGY (GJEMPS)   *
C*                                                                     *
C*                    REVISION DATE: 21 APRIL 2000                     *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* SUBROUTINE GJEMPS COMPUTES THE INVERSE OF AN N BY N MATRIX AND/OR   *
C* THE N SOLUTIONS OF A SET OF N LINEAR EQUATIONS IN N UNKNOWNS USING  *
C* GAUSS-JORDAN ELIMINATION WITH MAXIMUM PIVOT STRATEGY.               *
C*                                                                     *
C* INPUT VARIABLES:                                                    *
C*                                                                     *
C* EPS    = MINIMUM ALLOWABLE MAGNITUDE FOR A PIVOT ELEMENT (SUGGESTED *
C*          VALUE 1.0E-10)                                             *
C* INDIC  = CONTROL VARIABLE SUCH THAT:                                *
C*        = NEGATIVE, TO COMPUTE THE INVERSE OF N BY N MATRIX A IN     *
C*          PLACE                                                      *
C*        = ZERO    , TO COMPUTE THE N SOLUTIONS X(1)....X(N)          *
C*          CORRESPONDING TO THE SET OF LINEAR EQUATIONS WITH          *
C*          AUGMENTED MATRIX OF COEFFICIENTS IN THE N BY N+1 ARRAY A   *
C*          AND IN ADDITION COMPUTE THE INVERSE OF THE COEFFICIENT     *
C*          MATRIX IN PLACE                                            *
C*        = POSITIVE, TO COMPUTE THE N SOLUTIONS X(1)....X(N)          *
C*          CORRESPONDING TO THE SET OF LINEAR EQUATIONS WITH          *
C*          AUGMENTED MATRIX OF COEFFICIENTS IN THE N BY N+1 ARRAY A   *
C* N      = NUMBER OF ROWS IN MATRIX A                                 *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* DETER  = DETERMINATE OF THE ORIGINAL COEFFICIENT MATRIX FORMED BY   *
C*          THE FIRST N COLUMNS OF ARRAY A                             *
C* X      = VECTOR OF N SOLUTIONS                                      *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* A      = AUGMENTED COEFFICIENT MATRIX                               *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXNC   = MAXIMUM NUMBER OF COLUMNS (MXNC = MXNR + 1)                *
C* MXNCM1 = MAXIMUM NUMBER OF COLUMNS MINUS 1 (MXNCM1 = MXNC - 1)      *
C* MXNR   = MAXIMUM NUMBER OF ROWS                                     *
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)         *
C*                                                                     *
C* NOTE: SHOULD NO ACCEPTABLE PIVOT ELEMENT BE FOUND, COMPUTATIONS ARE *
C*       TERMINATED AND THE DETERMINATE IS RETURNED WITH A TRUE ZERO   *
C*       VALUE.                                                        *
C*                                                                     *
C* REF: CARNAHAN, B., LUTHER, H.A., AND WILKES, J.O.: APPLIED          *
C*      NUMERICAL METHODS, NEW YORK, 1969, JOHN WILEY & SONS, INC.     *
C*                                                                     *
C***********************************************************************
C
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      PARAMETER (MXNC=31,MXNCM1=30,MXNR=30,MXNRM1=29)
C
      DIMENSION A(MXNR,MXNC),IROW(MXNR),JCOL(MXNR),JORD(MXNR),X(MXNR),
     >Y(MXNR)
C
      COMMON/B/B(0:MXNRM1)
      COMMON/C/C(0:MXNRM1,0:MXNCM1)
C
      EQUIVALENCE (A(1,1),C(0,0)),(X(1),B(0))
C
C
C***********************************************************************
C*                                                                     *
C*                     BEGIN ELIMINATION PROCEDURE                     *
C*                                                                     *
C***********************************************************************
C
C
      IF(N.GT.MXNR) GO TO 114
      MAX=N
      IF(INDIC.GE.0) MAX=N+1
      DETER=1.0D+00
      DO 107 K=1,N
      KM1=K-1
C
C
C***********************************************************************
C*                                                                     *
C*                     SEARCH FOR THE PIVOT ELEMENT                    *
C*                                                                     *
C***********************************************************************
C
C
      PIVOT=0.0D+00
      DO 103 I=1,N
      DO 103 J=1,N
C
C
C***********************************************************************
C*                                                                     *
C*        SCAN IROW AND JCOL ARRAYS FOR INVALID PIVOT SUBSCRIPTS       *
C*                                                                     *
C***********************************************************************
C
C
      IF(K.EQ.1) GO TO 102
      DO 101 ISCAN=1,KM1
      DO 101 JSCAN=1,KM1
      IF(I.EQ.IROW(ISCAN)) GO TO 103
101   IF(J.EQ.JCOL(JSCAN)) GO TO 103
102   IF(DABS(A(I,J)).LE.DABS(PIVOT)) GO TO 103
      PIVOT=A(I,J)
      IROW(K)=I
      JCOL(K)=J
103   CONTINUE
C
C
```

```
C***********************************************************************00000121
C*                                                                     *00000122
C*             INSURE THE SELECTED PIVOT IS LARGER THAN EPS            *00000123
C*                                                                     *00000124
C***********************************************************************00000125
C                                                                       00000126
C                                                                       00000127
      IF(DABS(PIVOT).GT.EPS) GO TO 104                                  00000128
      DETER=0.0D+00                                                     00000129
      RETURN                                                            00000130
C                                                                       00000131
C                                                                       00000132
C***********************************************************************00000133
C*                                                                     *00000134
C*    UPDATE THE DETERMINATE VALUE AND NORMALIZE PIVOT ROW ELEMENTS    *00000135
C*                                                                     *00000136
C***********************************************************************00000137
C                                                                       00000138
C                                                                       00000139
104   DETER=DETER*PIVOT                                                 00000140
      DO 105 J=1,MAX                                                    00000141
105   A(IROW(K),J)=A(IROW(K),J)/PIVOT                                   00000142
C                                                                       00000143
C                                                                       00000144
C***********************************************************************00000145
C*                                                                     *00000146
C*              CARRY OUT ELIMINATION AND DEVELOP INVERSE              *00000147
C*                                                                     *00000148
C***********************************************************************00000149
C                                                                       00000150
C                                                                       00000151
      A(IROW(K),JCOL(K))=1.0D+00/PIVOT                                  00000152
      DO 107 I=1,N                                                      00000153
      AIJCK=A(I,JCOL(K))                                                00000154
      IF(I.EQ.IROW(K)) GO TO 107                                        00000155
      A(I,JCOL(K))=-AIJCK/PIVOT                                         00000156
      DO 106 J=1,MAX                                                    00000157
106   IF(J.NE.JCOL(K)) A(I,J)=A(I,J)-AIJCK*A(IROW(K),J)                 00000158
107   CONTINUE                                                          00000159
C                                                                       00000160
C                                                                       00000161
C***********************************************************************00000162
C*                                                                     *00000163
C*         ORDER SOLUTION VALUES (IF ANY) AND CREATE JORD ARRAY        *00000164
C*                                                                     *00000165
C***********************************************************************00000166
C                                                                       00000167
C                                                                       00000168
      DO 108 I=1,N                                                      00000169
      JORD(IROW(I))=JCOL(I)                                             00000170
108   IF(INDIC.GE.0) X(JCOL(I))=A(IROW(I),MAX)                          00000171
C                                                                       00000172
C                                                                       00000173
C***********************************************************************00000174
C*                                                                     *00000175
C*                      ADJUST SIGN OF DETERMINANT                     *00000176
C*                                                                     *00000177
C***********************************************************************00000178
C                                                                       00000179
C                                                                       00000180
```

```
      INTCH=0
      DO 109 I=1,N-1
      DO 109 J=I+1,N
      IF(JORD(J).GE.JORD(I)) GO TO 109
      JTEMP=JORD(J)
      JORD(J)=JORD(I)
      JORD(I)=JTEMP
      INTCH=INTCH+1
109   CONTINUE
      IF(INTCH/2*2.NE.INTCH) DETER=-DETER
C
C
C***********************************************************************
C*                                                                     *
C*                        UNSCRAMBLE THE INVERSE                       *
C*                                                                     *
C***********************************************************************
C
C
      IF(INDIC.GT.0) RETURN
      DO 111 J=1,N
      DO 110 I=1,N
110   Y(JCOL(I))=A(IROW(I),J)
      DO 111 I=1,N
111   A(I,J)=Y(I)
      DO 113 I=1,N
      DO 112 J=1,N
112   Y(IROW(J))=A(I,JCOL(J))
      DO 113 J=1,N
113   A(I,J)=Y(J)
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
114   WRITE(UNIT=6,FMT=201)N,MXNC
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,'ERROR IN SUBROUTINE GJEMPS: THE REQUESTED NUMBER OF ROWS
     >,N',/,'EXCEEDS THE DIMENSIONED MAXIMUM NUMBER OF ROWS, MXNR FOR',/
     >,'N =',I5,' MXNR =',I5,//,'REDIMENSION PARAMETERS:',//,'MXNR   = M
     >AXIMUM NUMBER OF ROWS',/,'MXNC   = MAXIMUM NUMBER OF COLUMNS',/,'M
     >XNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1',/,'MXNCM1 = MAXIMUM NUMBER
     > OF COLUMNS MINUS 1',//,'AND RECOMPILE.')
      END
      SUBROUTINE LSPRWC(*)
C
C
C***********************************************************************
C*                                                                     *
C*    LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS (LSPRWC)    *
C*                                                                     *
C*                     REVISION DATE: 6 JULY 2011                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*  SUBROUTINE LSPRWC PERFORMS A LEAST SQUARES POLYNOMIAL REGRESSION   *
C*  WITH CONSTRAINTS; THAT IS, A SET OF X-Y DATA POINTS IS CURVE FIT   *
C*  WITH AN NP ORDER POLYNOMIAL OF THE FORM                            *
C*                                                                     *
C*             P(X)=B0+B1*X+B2*X**2+B3*X**3+....+BNP*X**NP             *
C*                                                                     *
C*  WITH ANY POLYNOMIAL DERIVATIVES, ZERO THROUGH NP, SPECIFIED AT     *
C*  GIVEN X LOCATIONS.  THE PROCEDURE FOLLOWED IS THE METHOD OF LEAST  *
C*  SQUARES USING UNDETERMINED LAGRANGE MULTIPLIERS.                   *
C*                                                                     *
C* INPUT VARIABLES:                                                    *
C*                                                                     *
C* CV     = ARRAY OF POLYNOMIAL DERIVATIVE VALUES                      *
C* NC     = ARRAY OF POLYNOMIAL DERIVATIVE ORDERS (CONSTRAINT ORDER)   *
C* NCONST = NUMBER OF CONSTRAINTS OR (XC,NC,CV) DATA TRIPLES           *
C* NDATA  = NUMBER OF X-Y DATA PAIRS                                   *
C* NP     = POLYNOMIAL ORDER                                           *
C* X      = DATA POINT X VALUE                                         *
C* XC     = ARRAY OF POLYNOMIAL DERIVATIVE LOCATIONS (CONSTRAINT X     *
C*            VALUES)                                                  *
C* Y      = DATA POINT Y VALUE                                         *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* B      = ARRAY OF REGRESSION COEFFICIENTS                           *
C* SDEV   = STANDARD DEVIATION OF THE X-Y DATA POINTS ABOUT THE        *
C*            REGRESSION LINE                                          *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXANC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS                    *
C* MXNC   = MAXIMUM NUMBER OF COLUMNS (MXNC = MXNR + 1)                *
C* MXNCM1 = MAXIMUM NUMBER OF COLUMNS MINUS 1 (MXNCM1 = MXNC - 1)      *
C* MXNR   = MAXIMUM NUMBER OF ROWS                                     *
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)         *
C*                                                                     *
C* REF: CARNAHAN,B., LUTHER, H.A., AND WILKES, J.O.: APPLIED NUMERICAL *
C*      METHODS, NEW YORK, JOHN WILEY & SONS, 1969, PP. 571-584.       *
C*                                                                     *
C*      WEINSTOCK, R.: CALCULUS OF VARIATIONS WITH APPLICATIONS TO     *
C*      PHYSICS AND ENGINEERING, NEW YORK, DOVER PUBLICATIONS, INC.,   *
C*      1974, P. 6.                                                    *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
```

```
C
      INTEGER COL,COLMAX,FACTRL,ROW,ROWMAX
C
      PARAMETER (MXANC=20,MXNCM1=30,MXNRM1=29)
C
      COMMON/B/B(0:MXNRM1)
      COMMON/C/C(0:MXNRM1,0:MXNCM1)
      COMMON/CV/CV(MXANC)
      COMMON/NC/NC(MXANC)
      COMMON/NCONST/NCONST
      COMMON/NDATA/NDATA
      COMMON/NP/NP
      COMMON/SDEV/SDEV
      COMMON/XC/XC(MXANC)
C
C
C***********************************************************************
C*                                                                     *
C*           ZERO THE COEFFICIENT MATRIX AND SOLUTION VECTOR           *
C*                                                                     *
C***********************************************************************
C
C
      ROWMAX=NP+NCONST
      COLMAX=ROWMAX+1
      DO 101 ROW=0,ROWMAX
      DO 101 COL=0,COLMAX
101   C(ROW,COL)=0.0D+00
C
C
C***********************************************************************
C*                                                                     *
C*                  CALCULATE THE COEFFICIENT MATRIX                   *
C*                                                                     *
C***********************************************************************
C
C
      REWIND 1
C
C
C***********************************************************************
C*                                                                     *
C*          MINIMIZATION EQUATIONS - COEFFICIENTS OF THE B(J)          *
C*                                                                     *
C***********************************************************************
C
C
102   READ(UNIT=1,END=105,ERR=113)X,Y
      DO 104 JJ=0,NP
      ROW=JJ
      DO 103 J=0,NP
      COL=J
103   C(ROW,COL)=C(ROW,COL)+POWER(X,J+JJ)
C
C
C***********************************************************************
C*                                                                     *
C*                         VECTOR OF CONSTANTS                         *
C*                                                                     *
C***********************************************************************
```

```
C                                                                       00000121
C                                                                       00000122
104   C(ROW,COLMAX)=C(ROW,COLMAX)+Y*POWER(X,JJ)                         00000123
      GO TO 102                                                         00000124
C                                                                       00000125
C                                                                       00000126
C***********************************************************************00000127
C*                                                                     *00000128
C*                    COEFFICIENTS OF THE LAMBDA(L)                    *00000129
C*                                                                     *00000130
C***********************************************************************00000131
C                                                                       00000132
C                                                                       00000133
105   IF(NCONST.EQ.0) GO TO 108                                         00000134
      DO 106 JJ=0,NP                                                    00000135
      ROW=JJ                                                            00000136
      F1=FACTRL(JJ)                                                     00000137
      DO 106 L=1,NCONST                                                 00000138
      IF(JJ.LT.NC(L)) GO TO 106                                         00000139
      COL=NP+L                                                          00000140
      F2=F1/FACTRL(JJ-NC(L))/2.0D+00                                    00000141
      C(ROW,COL)=F2*POWER(XC(L),JJ-NC(L))                               00000142
106   CONTINUE                                                          00000143
C                                                                       00000144
C                                                                       00000145
C***********************************************************************00000146
C*                                                                     *00000147
C*           CONSTRAINT EQUATIONS - COEFFICIENTS OF THE B(J)           *00000148
C*                                                                     *00000149
C***********************************************************************00000150
C                                                                       00000151
C                                                                       00000152
      DO 107 L=1,NCONST                                                 00000153
      ROW=NP+L                                                          00000154
      C(ROW,COLMAX)=CV(L)                                               00000155
      DO 107 J=NC(L),NP                                                 00000156
      COL=J                                                             00000157
      F1=FACTRL(J)/FACTRL(J-NC(L))                                      00000158
107   C(ROW,COL)=F1*POWER(XC(L),J-NC(L))                                00000159
C                                                                       00000160
C                                                                       00000161
C***********************************************************************00000162
C*                                                                     *00000163
C*                     COMPUTE THE SOLUTION VECTOR                     *00000164
C*                                                                     *00000165
C***********************************************************************00000166
C                                                                       00000167
C                                                                       00000168
108   CALL GJEMPS(1.0D-20,+1,ROWMAX+1,DETER,*112)                       00000169
      IF(DETER.EQ.0.0D+00) GO TO 114                                    00000170
C                                                                       00000171
C                                                                       00000172
C***********************************************************************00000173
C*                                                                     *00000174
C*                   COMPUTE THE STANDARD DEVIATION                    *00000175
C*                                                                     *00000176
C***********************************************************************00000177
C                                                                       00000178
C                                                                       00000179
      SDEV=0.0D+00                                                      00000180
      IDENOM=NDATA-(NP+1)-NCONST-1
      IF(IDENOM.LE.0) RETURN
      DENOM=IDENOM
      REWIND 1
109   READ(UNIT=1,END=111,ERR=113)X,Y
      PX=0.0D+00
      DO 110 J=NP,1,-1
110   PX=(PX+B(J))*X
      PX=PX+B(0)
      SDEV=SDEV+(Y-PX)**2
      GO TO 109
111   SDEV=DSQRT(SDEV/DENOM)
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
112   WRITE(UNIT=6,FMT=201)
      RETURN 1
113   WRITE(UNIT=6,FMT=202)
      RETURN 1
114   WRITE(UNIT=6,FMT=203)
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT('SUBROUTINE GJEMPS WAS CALLED FROM SUBROUTINE LSPRWC')
202   FORMAT(/,'ERROR IN SUBROUTINE LSPRWC: UNFORMATTED READ ERROR ON UN
     >IT = 1')
203   FORMAT('SUBROUTINE GJEMPS WAS CALLED FROM SUBROUTINE LSPRWC')
      END
      SUBROUTINE PES(*)
C
C
C***********************************************************************
C*                                                                     *
C*                POLYNOMIAL EVALUATION SUBROUTINE (PES)               *
C*                                                                     *
C*                      REVISION DATE: 3 MAY 2002                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* GIVEN X AND THE COEFFICIENTS B(0),B(1),....,B(NP), SUBROUTINE PES   *
C* EVALUATES THE K-TH DERIVATIVE OF POLYNOMIALS OF THE FORM:           *
C*                                                                     *
C*             P(X)=B(0)+B(1)*X+B(2)*X**2+....+B(NP)*X**NP             *
C*                                                                     *
C* INPUT VARIABLES:                                                    *
C*                                                                     *
C* B      = ARRAY OF POLYNOMIAL COEFFICIENTS                           *
C* NP     = POLYNOMIAL ORDER                                           *
C* X      = X VALUE FOR EVALUATION OF THE POLYNOMIAL                   *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* R      = VALUE AT X OF THE RADIUS OF CURVATURE OF THE POLYNOMIAL    *
C* S      = VALUE AT X OF THE APPROXIMATE ARC LENGTH ALONG THE         *
C*          POLYNOMIAL                                                 *
C* YK     = VALUE AT X OF THE K-TH DERIVATIVE OF THE POLYNOMIAL        *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXNR   = MAXIMUM NUMBER OF ROWS                                     *
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)         *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      INTEGER FACTRL
C
      PARAMETER (MXNRM1=29)
C
      COMMON/B/B(0:MXNRM1)
      COMMON/NP/NP
C
      DIMENSION YK(0:9)
C
C
      REWIND 2
      REWIND 3
      S=0.0E+00
      N=0
101   READ(UNIT=2,END=106,ERR=107)X
      N=N+1
C
C
C***********************************************************************
```

```
C*                                                                     *
C*      EVALUATE THE NK-TH DERIVATIVE OF THE POLYNOMIAL (0 TO NP)      *
C*                                                                     *
C***********************************************************************
C
C
      DO 103 NK=0,NP
      YK(NK)=0.0D+00
      DO 102 J=NP,NK+1,-1
102   YK(NK)=(YK(NK)+FACTRL(J)*B(J)/FACTRL(J-NK))*X
103   YK(NK)=YK(NK)+FACTRL(NK)*B(NK)
C
C
C***********************************************************************
C*                                                                     *
C*         EVALUATE THE RADIUS OF CURVATURE OF THE POLYNOMIAL          *
C*                                                                     *
C***********************************************************************
C
C
      R=1.0E+32
      IF(NP.LT.2) GO TO 104
      IF(YK(2).EQ.0.0E+00) GO TO 104
      R=((1.0E+00+YK(1)*YK(1))**1.5)/DABS(YK(2))
104   CONTINUE
C
C
C***********************************************************************
C*                                                                     *
C*                      CALCULATE THE ARC LENGTH                       *
C*                                                                     *
C***********************************************************************
C
C
C
      IF(N.EQ.1) GO TO 105
      DeltaX=X-Xsave
      DeltaY=YK(0)-Ysave
      DeltaS=DSQRT(DeltaX*DeltaX+DeltaY*DeltaY)
      S=S+DeltaS
105   Xsave=X
      Ysave=YK(0)
      WRITE(UNIT=3,ERR=108)X,(YK(K),K=0,NP),R,S
      GO TO 101
C
C
106   END FILE 3
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
107   WRITE(UNIT=6,FMT=201)
      RETURN 1
108   WRITE(UNIT=6,FMT=202)
```

```
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                           FORMAT STATEMENTS                         *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,'ERROR IN SUBROUTINE PES: UNFORMATTED READ ERROR ON UNIT 
     >= 2')
202   FORMAT(/,'ERROR IN SUBROUTINE PES: UNFORMATTED WRITE ERROR ON UNIT
     > = 3')
      END
      SUBROUTINE PODS(*)
C
C
C***********************************************************************
C*                                                                     *
C*            POLYNOMIAL ORDER DEFINITION SUBROUTINE (PODS)            *
C*                                                                     *
C*                  REVISION DATE: 11 SEPTEMBER 2007                   *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*  SUBROUTINE PODS DEFINES THE POLYNOMIAL ORDER FOR PROGRAM LSPRWC.   *
C*                                                                     *
C* INPUT VARIABLES:                                                    *
C*                                                                     *
C* NDATA  = NUMBER OF X,Y DATA POINTS TO BE FIT                        *
C* NCONST = NUMBER OF CONSTRAINTS                                      *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* NP     = POLYNOMIAL ORDER                                           *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXNAC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS                    *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      PARAMETER (MXNAC=20,MXNR=30,MXNRM1=29)
C
      COMMON/NC/NC(MXNAC)
      COMMON/NCONST/NCONST
      COMMON/NDATA/NDATA
      COMMON/NP/NP
C
      DATA NP/3/
C
      NAMELIST/PARM/NP
C
C
C***********************************************************************
C*                                                                     *
C*                     INPUT THE POLYNOMIAL ORDER                      *
C*                                                                     *
C***********************************************************************
C
C
      NPMAX=MIN0(9,NDATA-NCONST-1)
101   WRITE(UNIT=6,FMT=201)NPMAX
      CALL NDWUS('PARM',6,*103)
      CALL NDWUSI('NP',1,1,1,NP,*103)
      CALL NDWUSE(*103)
      CALL NDRUS('PARM',4,*101,*104)
      READ(UNIT=4,NML=PARM,ERR=101)
      CALL NDRUSE(*101,*104)
      IF(NP.LT.0.OR.NP.GT.9) GO TO 105
      IF(NP.GT.NPMAX) GO TO 106
      DO 102 I=1,NCONST
      IF(NC(I).LT.0.OR.NC(I).GT.NP) GO TO 107
102   CONTINUE
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
103   WRITE(UNIT=6,FMT=204)
      GO TO 108
104   WRITE(UNIT=6,FMT=205)
      GO TO 108
105   WRITE(UNIT=6,FMT=206)NP
      GO TO 108
106   WRITE(UNIT=6,FMT=207)NP,NCONST,NDATA,NPMAX
      GO TO 108
107   WRITE(UNIT=6,FMT=208)NC(I),I,NP
108   WRITE(UNIT=6,FMT=202)
      READ(UNIT=5,FMT=203)CCV
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,'INPUT THE POLYNOMIAL ORDER BY NAMELIST WHERE:',//,'NP
     > = THE POLYNOMIAL ORDER',//,'NOTE: NP MUST BE IN THE RANGE OF 0 TO
     > 9.',/,T7,'THERE MUST BE AT LEAST NP+NCONST+1 X-Y DATA POINTS.',/,
     >T7,'THE MAXIMUM VALUE FOR NP IS ',I5,'.',//,'CURRENT VALUES ARE:')
202   FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')
203   FORMAT(A1)
204   FORMAT('SUBROUTINE NDWUS WAS CALLED FROM SUBROUTINE PODS')
205   FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE PODS')
206   FORMAT(/,'ERROR IN SUBROUTINE PODS: NP = ',I5,' IS NOT ALLOWED.',/
     >,'THE POLYNOMIAL ORDER MUST BE GREATER THAN OR EQUAL TO 0 AND LESS
     > THAN OR EQUAL',/,'TO 9.')
207   FORMAT(/,'ERROR IN SUBROUTINE PODS: NP = ',I5,' IS NOT ALLOWED FOR
     > NCONST = ',I5,' AND',/,'NDATA = ',I5,'.  THERE MUST BE AT LEAST N
     >P+2*NCONST X-Y DATA POINTS.  THE',/,'MAXIMUM VALUE FOR NP IS ',I5,
     >'.')
208   FORMAT(/,'ERROR IN SUBROUTINE PODS: NC(',I3,') = ',I5,' IS NOT ALL
     >OWED FOR NP = ',I5,'.',/,'THE CONSTRAINT ORDER MUST BE GREATER THA
     >N OR EQUAL TO 0 AND LESS THAN OR EQUAL TO NP.')
      END
      FUNCTION POWER(X,N)
C
C
C***********************************************************************
C*                                                                     *
C*     INTEGER POWERS OF REAL NUMBERS FUNCTION SUBPROGRAM (POWER)     *
C*                                                                     *
C*                     REVISION DATE: 15 JUNE 1989                     *
C*                                                                     *
C***********************************************************************
C
C
      REAL*8 POWER,X
      POWER=1.0D+00
      IF(N.EQ.0) RETURN
      DO 101 I=1,N
101   POWER=POWER*X
      END
      SUBROUTINE QPS(*)
C
C
C***********************************************************************
C*                                                                     *
C*                     QUICK PLOT SUBROUTINE (QPS)                     *
C*                                                                     *
C*                     REVISION DATE: 1 JULY 2011                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*    SUBROUTINE QPS WRITES DATA FILES TO CONSTRUCT A QUICK PLOT OF    *
C*    RESULTS  FOR  PROGRAM  LSPRWC USING THE  PostScript CARTESIAN    *
C*    COORDINATE PLOT PROGRAM PSCCPP.                                  *
C*                                                                     *
C* INPUT VARIABLES:                                                    *
C*                                                                     *
C* NCONST = NUMBER OF CONSTRAINTS                                      *
C* X      = X VALUE FOR EVALUATION OF THE POLYNOMIAL                   *
C* XC     = ARRAY OF CONSTRAINT X VALUES                               *
C* Y      = Y VALUE FROM EVALUATION OF THE POLYNOMIAL AT X             *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MXNAC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS                    *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER*10 File
C
      PARAMETER (MXNAC=20,One=1.0E+00,Zero=0.0E+00)
C
      COMMON/NCONST/NCONST
      COMMON/XC/XC(MXNAC)
C
      DIMENSION File(3)
C
      DATA File/'@lsprwc.cl','@lsprwc.ed','@lsprwc.dp'/,Norm/0/
C
      NAMELIST/PARM/Norm,Xmax,Xmin,Ymax,Ymin
C
      FNX(Q)=(Q-Xmin)/(Xmax-Xmin)
      FNY(Q)=(Q-Ymin)/(Ymax-Ymin)
C
C
C***********************************************************************
C*                                                                     *
C*                     FIND THE RANGE OF THE DATA                      *
C*                                                                     *
C***********************************************************************
C
C
      Xmin=+1.0D+75
      Xmax=-1.0D+75
      Ymin=+1.0D+75
      Ymax=-1.0D+75
      REWIND 1
101   READ(UNIT=1,END=102,ERR=116)X,Y
      Xmin=DMIN1(X,Xmin)
      Xmax=DMAX1(X,Xmax)
      Ymin=DMIN1(Y,Ymin)
      Ymax=DMAX1(Y,Ymax)
      GO TO 101
102   REWIND 3
103   READ(UNIT=3,END=104,ERR=117)X,Y
      Xmin=DMIN1(X,Xmin)
      Xmax=DMAX1(X,Xmax)
      Ymin=DMIN1(Y,Ymin)
      Ymax=DMAX1(Y,Ymax)
      GO TO 103
104   IF(NCONST.EQ.0) GO TO 106
      DO 105 I=1,NCONST
      Xmin=DMIN1(XC(I),Xmin)
105   Xmax=DMAX1(XC(I),Xmax)
106   IF(Xmin.GE.Xmax) RETURN
      IF(Ymin.GE.Ymax) RETURN
107   WRITE(UNIT=6,FMT=201)
      CALL NDWUS('PARM',6,*118)
      CALL NDWUSI('Norm',1,1,1,Norm,*118)
      CALL NDWUSD('Xmax',1,1,1,Xmax,*118)
      CALL NDWUSD('Xmin',1,1,1,Xmin,*118)
      CALL NDWUSD('Ymax',1,1,1,Ymax,*118)
      CALL NDWUSD('Ymin',1,1,1,Ymin,*118)
      CALL NDWUSE(*118)
      CALL NDRUS('PARM',4,*107,*119)
      READ(UNIT=4,NML=PARM,ERR=107)
      CALL NDRUSE(*107,*119)
C
C
C***********************************************************************
C*                                                                     *
C*                    MARK THE CONSTRAINT LOCATIONS                    *
C*                                                                     *
C***********************************************************************
C
C
      N=1
      OPEN(UNIT=4,FILE=File(N),STATUS='UNKNOWN',FORM='FORMATTED',
     >ERR=120)
      IF(NCONST.EQ.0) GO TO 109
      DO 108 I=1,NCONST
      IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)XC(I),Ymin,INT(Zero)
      IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(XC(I)),Zero,
     >INT(Zero)
      IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)XC(I),Ymax,INT(One)
      IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(XC(I)),One,
     >INT(One)
108   CONTINUE
      GO TO 110
109   WRITE(UNIT=4,FMT=202,ERR=121)Zero,Zero,INT(Zero)
110   CLOSE(UNIT=4,STATUS='KEEP',ERR=122)
C
C
C***********************************************************************
C*                                                                     *
```

```
C*                      WRITE THE EVALUATED DATA                       *
C*                                                                     *
C***********************************************************************
C
C
      N=2
      OPEN(UNIT=4,FILE=File(N),STATUS='UNKNOWN',FORM='FORMATTED',
     >ERR=120)
111   REWIND 3
112   READ(UNIT=3,END=113,ERR=117)X,Y
      IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)X,Y
      IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(X),FNY(Y)
      GO TO 112
113   CLOSE(UNIT=4,STATUS='KEEP',ERR=122)
C
C
C***********************************************************************
C*                                                                     *
C*                MARK THE X-Y DATA POINTS TO BE FITTED                *
C*                                                                     *
C***********************************************************************
C
C
      N=3
      OPEN(UNIT=4,FILE=File(N),STATUS='UNKNOWN',FORM='FORMATTED',
     >ERR=120)
      REWIND 1
114   READ(UNIT=1,END=115,ERR=116)X,Y
      IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)X,Y
      IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(X),FNY(Y)
      GO TO 114
115   CLOSE(UNIT=4,STATUS='KEEP',ERR=122)
C
C
C***********************************************************************
C*                                                                     *
C*                         CONTSTRUCT THE PLOT                         *
C*                                                                     *
C***********************************************************************
C
C
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
116   WRITE(UNIT=6,FMT=205)
      RETURN 1
117   WRITE(UNIT=6,FMT=206)
      RETURN 1
118   WRITE(UNIT=6,FMT=207)
      RETURN 1
119   WRITE(UNIT=6,FMT=208)
      RETURN 1
120   WRITE(UNIT=6,FMT=209)File(N)
```

```
      RETURN 1
121   WRITE(UNIT=6,FMT=210)File(N)
      RETURN 1
122   WRITE(UNIT=6,FMT=211)File(N)
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,'INPUT THE PLOT VARIABLES BY NAMELIST WHERE:',//,'Norm
     >= RENormILIZATION CONTROL',/,T8,'= 0, TO NOT NormALIZE THE PLOT DA
     >TA',/,T8,'= 1, TO NormALIZE THE PLOT DATA',/,'Xmax   = MAXIMUM X-V
     >ALUE',/,'Xmin   = MINIMUM X-VALUE',/,'Ymax   = MAXIMUM Y-VALUE',/,
     >'Ymin   = MINIMUM Y-VALUE',//,'CURRENT VALUES ARE:')
202   FORMAT(2E15.6,I15)
203   FORMAT('lsprwc.l')
204   FORMAT('lsprwc.ps')
205   FORMAT(/,'ERROR IN SUBROUTINE QPS: UNFORMATTED READ ERROR ON UNIT1
     >')
206   FORMAT(/,'ERROR IN SUBROUTINE QPS: UNFORMATTED READ ERROR ON UNIT3
     >')
207   FORMAT('SUBROUTINE NDWUS WAS CALLED FROM SUBROUTINE QPS')
208   FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE QPS')
209   FORMAT(/,'ERROR IN SUBROUTINE QPS: OPEN ERROR ON UNIT 4, STATUS = 
     >"UNKNOWN", FILE =',A10)
210   FORMAT(/,'ERROR IN SUBROUTINE QPS: WRITE ERROR ON UNIT 4, FILE = '
     >,A10)
211   FORMAT(/,'ERROR IN SUBROUTINE QPS: CLOSE ERROR ON UNIT 4, FILE =',
     >A10)
      END
      SUBROUTINE XDPEDS(LCV2,*,*)
C
C
C***********************************************************************
C*                                                                     *
C*     X DATA POINTS FOR EVALUATION DEFINITION SUBROUTINE (XDPEDS)     *
C*                                                                     *
C*                     REVISION DATE: 1 JULY 2011                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*  SUBROUTINE XDPEDS DEFINES THE X COORDINATE SET FOR EVALUATION OF   *
C*  THE RESULTANT LEAST SQUARES POLYNOMIAL SET.                        *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* X      = ARRAY OF X COORDINATES FOR EVALUATION OF THE LEAST SQUARES *
C*          POLYNOMIAL SET                                             *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* NEVAL  = NUMBER OF X COORDINATES FOR EVALUATION                     *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MNXCE  = MAXIMUM NUMBER OF X COORDINATES FOR POLYNOMIAL EVALUATION  *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER CCV*1,DFmt*3,File*80,Fmt*80
C
      PARAMETER (R_NaN=-999.E+00)
      PARAMETER (MNXCE=101)
C
      COMMON/NEVAL/NEVAL
      COMMON/XINT/XINT
      COMMON/XMAX/XMAX
      COMMON/XMIN/XMIN
C
      DIMENSION X(MNXCE)
C
      DATA DFmt/'(*)'/,NEVAL/0/
C
      NAMELIST/PARM/X,XINT,XMAX,XMIN
C
C
C***********************************************************************
C*                                                                     *
C*                INPUT THE X DATA POINTS FOR EVALUATION               *
C*                   OF THE LEAST SQUARES POLYNOMIAL                   *
C*                                                                     *
C***********************************************************************
C
C
      IF(NEVAL.EQ.0) WRITE(UNIT=6,FMT=201)
      IF(NEVAL.NE.0) WRITE(UNIT=6,FMT=202)
      READ(UNIT=5,FMT=203)LCV1
      REWIND 2
      LCV2=1
      GO TO (101,107,112,116),LCV1
C
C
C***********************************************************************
C*                                                                     *
C*                           NAMELIST INPUT                            *
C*                                                                     *
C***********************************************************************
C
C
101   DO 102 I=1,MNXCE
102   X(I)=R_NaN
103   WRITE(UNIT=6,FMT=204)MNXCE
      WRITE(UNIT=6,FMT=205)
      CALL NDRUS('PARM',4,*103,*117)
      READ(UNIT=4,NML=PARM,ERR=103)
      CALL NDRUSE(*103,*117)
      DO 104 NEVAL=MNXCE,1,-1
      IF(X(NEVAL).NE.R_NaN) GO TO 105
104   CONTINUE
105   DO 106 I=1,NEVAL
106   WRITE(UNIT=2,ERR=122)X(I)
      GO TO 115
C
C
C***********************************************************************
C*                                                                     *
C*                      FORMATTED DISK FILE INPUT                      *
C*                                                                     *
C***********************************************************************
C
C
107   WRITE(UNIT=6,FMT=206)
      WRITE(UNIT=6,FMT=207)
      READ(UNIT=5,FMT=208)File
108   WRITE(UNIT=6,FMT=209)
      WRITE(UNIT=6,FMT=210)
      CALL DFCUS(FMT,DFmt,*108)
      OPEN(UNIT=4,FILE=File,STATUS='OLD',ERR=118)
      REWIND 4
      NEVAL=0
      IF(Fmt(1:3).EQ.DFmt) GO TO 110
109   READ(UNIT=4,FMT=Fmt,END=111,ERR=119)X(1)
      WRITE(UNIT=2,ERR=122)X(1)
      NEVAL=NEVAL+1
      GO TO 109
110   READ(UNIT=4,FMT=*,END=111,ERR=119)X(1)
      WRITE(UNIT=2,ERR=122)X(1)
      NEVAL=NEVAL+1
      GO TO 110
111   CLOSE(UNIT=4,STATUS='KEEP',ERR=120)
      GO TO 115
C
C
C***********************************************************************
C*                                                                     *
```

```
C*                MINIMUM, MAXIMUM, AND INTERVAL VALUES                *
C*                                                                     *
C***********************************************************************
C
C
112   WRITE(UNIT=6,FMT=211)
      CALL NDWUS('PARM',6,*121)
      CALL NDWUSD('XINT',1,1,1,XINT,*121)
      CALL NDWUSD('XMAX',1,1,1,XMAX,*121)
      CALL NDWUSD('XMIN',1,1,1,XMIN,*121)
      CALL NDWUSE(*121)
      CALL NDRUS('PARM',4,*112,*117)
      READ(UNIT=4,NML=PARM,ERR=112)
      CALL NDRUSE(*112,*117)
      NEVAL=0
      X(1)=XMIN
      DUM1=XMAX+XINT/2
113   IF(X(1).GT.DUM1) GO TO 114
      WRITE(UNIT=2,ERR=122)X(1)
      NEVAL=NEVAL+1
      DUM3=X(1)
      X(1)=XMIN+NEVAL*XINT
      GO TO 113
114   IF(DUM3.EQ.XMAX) GO TO 115
      WRITE(UNIT=2,ERR=122)XMAX
      NEVAL=NEVAL+1
115   END FILE 2
116   RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
117   WRITE(UNIT=6,FMT=212)
      GO TO 124
118   WRITE(UNIT=6,FMT=213)
      CALL CBUS06(6,File,*123)
      RETURN 1
119   WRITE(UNIT=6,FMT=214)
      CALL CBUS06(6,File,*123)
      RETURN 1
120   WRITE(UNIT=6,FMT=215)
      CALL CBUS06(6,File,*123)
      RETURN 1
121   WRITE(UNIT=6,FMT=216)
      GO TO 124
122   WRITE(UNIT=6,FMT=217)
      RETURN 1
123   WRITE(UNIT=6,FMT=218)
      RETURN 1
124   WRITE(UNIT=6,FMT=219)
      READ(UNIT=5,FMT=220)CCV
      RETURN 2
C
C
C***********************************************************************
```

```
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATE
     >D FROM THE FOLLOWING',/,'LIST:',//,'1, FOR KEYBOARD INPUT VIA NAME
     >LIST',/,'2, FOR FORMATTED DISK FILE INPUT',/,'3, FOR INPUT OF MINI
     >MUM, MAXIMUM, AND INTERVAL VALUES')
202   FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATE
     >D FROM THE FOLLOWING',/,'LIST:',//,'1, FOR KEYBOARD INPUT VIA NAME
     >LIST',/,'2, FOR FORMATTED DISK FILE INPUT',/,'3, FOR INPUT OF MINI
     >MUM, MAXIMUM, AND INTERVAL VALUES',/,'4, TO USE THE PREVIOUSLY PRE
     >SCRIBED X VALUES')
203   FORMAT(I1)
204   FORMAT(/,'INPUT THE DATA SET X VALUES (',I3,' MAX) BY NAMELIST WHE
     >RE:',//,'X      = ARRAY OF X VALUES',//,'CURRENT VALUES ARE:')
205   FORMAT(/,'$PARM X=____,____,____$',/)
206   FORMAT(/,'DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS X VALUE
     >S WHERE:',//,'X      = X VALUE')
207   FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
     >')
208   FORMAT(A80)
209   FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)')
210   FORMAT('NOTE: AN EXAMPLE FORMAT IS "(E15.6)"',/,T7,'ENTER "(*)" FO
     >R A FREE FIELD READ (DEFAULT FORMAT)')
211   FORMAT(/,'INPUT THE X DATA MAXIMUM, MINIMUM, AND INTERVAL VALUES B
     >Y NAMELIST WHERE:',//,'XINT   = X DATA POINT INTERVAL',/,'XMAX   =
     > MAXIMUM X DATA VALUE',/,'XMIN   = MINIMUM X DATA VALUE',//,'CURRE
     >NT VALUES ARE:')
212   FORMAT('SUBROUTINE NDRUS WAS CALLED FROM PROGRAM LSPRWC')
213   FORMAT(/,'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 4, STATUS =
     >"OLD", FILE =',/)
214   FORMAT(/,'ERROR IN PROGRAM LSPRWC: READ ERROR ON UNIT 4, FILE =',/
     >)
215   FORMAT(/,'ERROR IN PROGRAM LSPRWC: CLOSE ERROR ON UNIT 4, FILE =',
     >/)
216   FORMAT('SUBROUTINE NDWUS WAS CALLED FROM PROGRAM LSPRWC')
217   FORMAT(/,'ERROR IN PROGRAM LSPRWC: UNFORMATTED WRITE ERROR ON UNIT
     > 2')
218   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM PROGRAM LSPRWC')
219   FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')
220   FORMAT(A1)
      END
      SUBROUTINE XYDIS(*,*)
C
C
C***********************************************************************
C*                                                                     *
C*                  X-Y DATA INPUT SUBROUTINE (XYDIS)                  *
C*                                                                     *
C*                     REVISION DATE: 1 JULY 2011                      *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* SUBROUTINE XYDIS DEFINES THE X,Y DATA POINTS TO BE FIT FOR PROGRAM *
C* LSPRWC.                                                             *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* X      = ARRAY OF X COORDINATES                                     *
C* Y      = ARRAY OF Y COORDINATES                                     *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* NDATA  = NUMBER OF X,Y DATA POINTS TO BE FIT                        *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* MNXYDP = MAXIMUM NUMBER OF (X,Y) DATA PAIRS FOR NAMELIST INPUT      *
C*                                                                     *
C***********************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CHARACTER CCV*1,Crdimg*80,DFmt*3,File*80,Fmt*80,Fmt_s*80
C
      PARAMETER (R_NaN=-999.E+00)
      PARAMETER (MNXYDP=101)
C
      COMMON/NDATA/NDATA
      COMMON/XINT/XINT
      COMMON/XMAX/XMAX
      COMMON/XMIN/XMIN
C
      DIMENSION X(MNXYDP),Y(MNXYDP)
C
      DATA DFmt/'(*)'/
C
      NAMELIST/PARM/X,Y
C
C
C***********************************************************************
C*                                                                     *
C*                INPUT THE X-Y DATA POINTS TO BE FITTED               *
C*                                                                     *
C***********************************************************************
C
C
      WRITE(UNIT=6,FMT=203)
      READ(UNIT=5,FMT=204)LCV1
      REWIND 1
      IF(LCV1.EQ.2) GO TO 106
C
C
C***********************************************************************
C*                                                                     *
C*                           NAMELIST INPUT                            *
C*                                                                     *
C***********************************************************************
C
C
      DO 101 I=1,MNXYDP
      X(I)=R_NaN
101   Y(I)=0.0
102   WRITE(UNIT=6,FMT=205)MNXYDP
      WRITE(UNIT=6,FMT=206)
      CALL NDRUS('PARM',4,*102,*119)
      READ(UNIT=4,NML=PARM,ERR=102)
      CALL NDRUSE(*102,*119)
      DO 103 NDATA=MNXYDP,1,-1
      IF(X(NDATA).NE.R_NaN) GO TO 104
103   CONTINUE
104   DO 105 I=1,NDATA
105   WRITE(UNIT=1,ERR=120)X(I),Y(I)
      GO TO 112
C
C
C***********************************************************************
C*                                                                     *
C*                      FORMATTED DISK FILE INPUT                      *
C*                                                                     *
C***********************************************************************
C
C
106   WRITE(UNIT=6,FMT=207)
      WRITE(UNIT=6,FMT=208)
      READ(UNIT=5,FMT=209)File
107   WRITE(UNIT=6,FMT=210)
      WRITE(UNIT=6,FMT=211)
      CALL DFCUS(Fmt,DFmt,*107)
C
      OPEN(UNIT=4,FILE=File,STATUS='OLD',ERR=121)
      REWIND 4
      NDATA=0
C
108   READ(UNIT=4,FMT=209,END=122,ERR=123)Crdimg
      CALL CBUS01(Crdimg,L)
      IF(L.EQ.0) GO TO 108
      IF(Fmt(1:3).NE.DFmt) READ(Crdimg,FMT=Fmt,ERR=108)X(1),Y(1)
      IF(Fmt(1:3).EQ.DFmt) READ(Crdimg,*,ERR=108)X(1),Y(1)
      BACKSPACE 4
C
      IF(Fmt(1:3).EQ.DFmt) GO TO 110
109   READ(UNIT=4,FMT=Fmt,END=111,ERR=123)X(1),Y(1)
      WRITE(UNIT=1,ERR=120)X(1),Y(1)
      NDATA=NDATA+1
      GO TO 109
110   READ(UNIT=4,FMT=*,END=111,ERR=123)X(1),Y(1)
      WRITE(UNIT=1,ERR=120)X(1),Y(1)
      NDATA=NDATA+1
      GO TO 110
111   CLOSE(UNIT=4,STATUS='KEEP',ERR=125)
112   END FILE 1
C
C
C***********************************************************************
C*                                                                     *
C*   SEARCH FOR THE MINIMUM AND MAXIMUM X VALUES WITHIN THE DATA SET   *
C*                                                                     *
C***********************************************************************
C
C
      REWIND 1
      XMAX=-1.0D+75
      XMIN=+1.0D+75
      XSAV=-1.0D+75
113   READ(UNIT=1,END=114,ERR=124)X(1),Y(1)
      IF(X(1).LT.XSAV) GO TO 126
      XMAX=DMAX1(XMAX,X(1))
      XMIN=DMIN1(XMIN,X(1))
      XSAV=X(1)
      GO TO 113
114   IF(XMAX.EQ.XMIN) GO TO 127
      XINT=(XMAX-XMIN)/1.0D+02
C
C
C***********************************************************************
C*                                                                     *
C*                      DISPLAY THE X-Y DATA SET                       *
C*                                                                     *
C***********************************************************************
C
C
115   WRITE(UNIT=6,FMT=212)
      CALL YNOUS(*116,*118,*115)
116   REWIND 1
      WRITE(UNIT=6,FMT=213)
      DO 117 I=1,NDATA
      READ(UNIT=1,ERR=124)X(1),Y(1)
      IF(MOD(I,20).NE.0) GO TO 117
      WRITE(UNIT=6,FMT=201)
      READ(UNIT=5,FMT=202)CCV
      WRITE(UNIT=6,FMT=213)
117   WRITE(UNIT=6,FMT=214)I,X(1),Y(1)
      WRITE(UNIT=6,FMT=201)
      READ(UNIT=5,FMT=202)CCV
118   RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                            ERROR MESSAGES                           *
C*                                                                     *
C***********************************************************************
C
C
119   WRITE(UNIT=6,FMT=215)
      GO TO 129
120   WRITE(UNIT=6,FMT=216)
      RETURN 1
```

```
121   WRITE(UNIT=6,FMT=217)
      CALL CBUS06(6,File,*128)
      RETURN 1
122   WRITE(UNIT=6,FMT=218)
      CALL CBUS06(6,File,*128)
      RETURN 1
123   WRITE(UNIT=6,FMT=219)
      CALL CBUS06(6,File,*128)
      RETURN 1
124   WRITE(UNIT=6,FMT=220)
      RETURN 1
125   WRITE(UNIT=6,FMT=221)
      CALL CBUS06(6,File,*128)
      RETURN 1
126   WRITE(UNIT=6,FMT=222)
      GO TO 129
127   WRITE(UNIT=6,FMT=223)
      GO TO 129
128   WRITE(UNIT=6,FMT=224)
      RETURN 1
129   WRITE(UNIT=6,FMT=201)
      READ(UNIT=5,FMT=202)CCV
      RETURN 2
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')
202   FORMAT(A1)
203   FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE X-Y DATA TO BE FITTED
     > FROM',/,'THE FOLLOWING LIST:',//,'1, FOR KEYBOARD INPUT VIA NAMEL
     >IST',/,'2, FOR FORMATTED DISK FILE INPUT',//,'NOTE: THE X-Y DATA M
     >UST BE MONATONICALLY INCREASING IN X.')
204   FORMAT(I1)
205   FORMAT(/,'INPUT THE DATA SET (X,Y) PAIRS (',I3,' MAX) BY NAMELIST
     >WHERE:',//,'X      = ARRAY OF X VALUES',/,'Y      = ARRAY OF Y VAL
     >UES',//,'CURRENT VALUES ARE:')
206   FORMAT(/,'$PARM X=____,____,____,Y=____,____,____$',/)
207   FORMAT(/,'DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS (X,Y) P
     >AIRS WHERE:',//,'X      = X VALUE',/,'Y      = Y VALUE')
208   FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
     >')
209   FORMAT(A80)
210   FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)')
211   FORMAT('NOTE: AN EXAMPLE FORMAT IS "(2E15.6)"',/,T7,'ENTER"(*)" FO
     >R A FREE FIELD READ (DEFAULT FORMAT)')
212   FORMAT(/,'SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
     >')
213   FORMAT(/,T10,'NO.',T26,'X',T49,'Y')
214   FORMAT(T8,I5,1P2D23.13)
215   FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE XYDIS')
216   FORMAT(/,'ERROR IN SUBROUTINE XYDIS: UNFORMATTED WRITE ERROR ON UN
     >IT 1')
217   FORMAT(/,'ERROR IN SUBROUTINE XYDIS: OPEN ERROR ON UNIT 4, STATUS
     >= "OLD", FILE =',/)
```

```
218   FORMAT(/,'ERROR IN SUBROUTINE XYDIS:  UNANTICPATED EOF ON UNIT 4,
     >FILE =',/)
219   FORMAT(/,'ERROR IN SUBROUTINE XYDIS: READ ERROR ON UNIT 4, FILE ='
     >,/)
220   FORMAT(/,'ERROR IN SUBROUTINE XYDIS: UNFORMATTED READ ERROR ON UNI
     >T 1')
221   FORMAT(/,'ERROR IN PROGRAM LSPRWC: CLOSE ERROR ON UNIT 4, FILE =',
     >/)
222   FORMAT(/,'ERROR IN SUBROUTINE XYDIS: ALL X-Y DATA POINTS MUST BE',
     >/,'MONOTONICALLY INCREASING IN X.')
223   FORMAT(/,'ERROR IN SUBROUTINE XYDIS: ALL X-Y DATA POINTS HAVE THE
     >SAME X VALUE LEADING TO',/,'A SINGULAR MATRIX.')
224   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE XYDIS')
      END
```

APPENDIX C
FORTRAN TOOL LIBRARY

```
      SUBROUTINE CBUS01(Buf,L)
C
C
C***********************************************************************
C*                                                                     *
C*            CHARACTER BUFFER UTILITY SUBROUTINE (CBUS01)             *
C*                                                                     *
C*                   REVISION DATE: 3 FEBRUARY 1997                    *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* SUBROUTINE CBUS01 FINDS THE NON-TRAILING-BLANK LENGTH OF CHARACTER  *
C* BUFFERS.                                                            *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* Buf    = CHARACTER BUFFER                                           *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* L      = NON-TRAILING-BLANK LENGTH OF Buf                           *
C*                                                                     *
C***********************************************************************
C
C
      CHARACTER Blank*1,Buf*(*)
C
      SAVE Blank
C
      DATA Blank/' '/
C
      DO 101 L=LEN(Buf),1,-1
      IF(Buf(L:L).NE.Blank) RETURN
101   CONTINUE
      L=0
      END
      SUBROUTINE CBUS06(LUnit,Buf,*)
C
C
C***********************************************************************
C*                                                                     *
C*            CHARACTER BUFFER UTILITY SUBROUTINE (CBUS06)             *
C*                                                                     *
C*                    REVISION DATE: 11 MARCH 1999                     *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*    SUBROUTINE CBUS06 WRITES CHARACTER BUFFERS, LESS THE TRAILING    *
C*    BLANKS, TO UNIT LUnit USING DYNAMIC FORMAT CONSTRUCTION.         *
C*                                                                     *
C* INPUT/OUTPUT VARIABLES:                                             *
C*                                                                     *
C* LUnit  = LOGICAL UNIT NUMBER                                        *
C* Buf    = CHARACTER BUFFER                                           *
C*                                                                     *
C***********************************************************************
C
C
      CHARACTER Buf*(*),Fmt*15
C
      INTEGER LUnit
C
      CALL CBUS01(Buf,L)
      IF(L.EQ.0) L=1
      WRITE(Fmt,FMT=201,ERR=101)L
      WRITE(UNIT=LUnit,FMT=Fmt,ERR=102)(Buf(1:L))
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
101   WRITE(UNIT=6,FMT=202)
      RETURN 1
102   WRITE(UNIT=6,FMT=203)LUnit
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(2H(A,I5,1H))
202   FORMAT(/,'ERROR IN SUBROUTINE CBUS06: WRITE ERROR',/,'WHILE ATTEMP
     >TING TO WRITE VARIABLE L TO Fmt')
203   FORMAT(/,'ERROR IN SUBROUTINE CBUS06: WRITE ERROR',/,'WHILE ATTEMP
     >TING TO WRITE VARIABLE Buf TO UNIT = ',I2)
```

```
END                                                               00000061
```

```
      SUBROUTINE DFCUS(Fmt,DFmt,*)
C
C
C***********************************************************************
C*                                                                     *
C*       DYNAMIC FORMAT CONSTRUCTION UTILITY SUBROUTINE (DFCUS)        *
C*                                                                     *
C*                   REVISION DATE: 20 NOVEMBER 2000                   *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*   SUBROUTINE DFCUS READS A FORTRAN FORMAT FROM UNIT 5, STORES THE   *
C*   FORMAT IN A CHARACTER BUFFER, AND CHECKS THE FORMAT FOR ERRORS.   *
C*   Blank FORMATS ARE REPLACED WITH A DEFAULT FORMAT.                 *
C*                                                                     *
C* INPUT VARIABLES:                                                    *
C*                                                                     *
C* DFmt   = DEFAULT FORMAT CHARACTER BUFFER                            *
C*                                                                     *
C* OUTPUT VARIABLES:                                                   *
C*                                                                     *
C* Fmt    = FORMAT CHARACTER BUFFER                                    *
C*                                                                     *
C***********************************************************************
C
C
      CHARACTER CCV*1,DFmt*(*),Fmt*(*),LParn*1,RParn*1
C
      SAVE LParn,RParn
C
      DATA LParn/'('/,RParn/')'/
C
      READ(UNIT=5,FMT=201)Fmt
      ENTRY DFCUSE(Fmt,DFmt,*)
      CALL CBUS03(Fmt)
      CALL CBUS01(Fmt,LFmt)
      IF(LFmt.GT.0) GO TO 101
      CALL CBUS01(DFmt,LDFmt)
      LFmt=LEN(Fmt)
      IF(LDFmt.GT.LFmt) GO TO 102
      Fmt(1:LDFmt)=DFmt(1:LDFmt)
      RETURN
101   IF(Fmt(1:1).EQ.LParn.AND.Fmt(LFmt:LFmt).EQ.RParn) RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
      WRITE(UNIT=6,FMT=202)
      CALL CBUS06(6,Fmt,*103)
      WRITE(UNIT=6,FMT=203)
      READ(UNIT=5,FMT=201)CCV
      RETURN 1
102   WRITE(UNIT=6,FMT=204)LDFmt,LFmt
```

```
      STOP
103   WRITE(UNIT=6,FMT=205)
      RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(A80)
202   FORMAT(/,'ERROR IN SUBROUTINE DFCUS: ILLEGAL VALUE FOR Fmt =')
203   FORMAT('FORMATS MUST BEGIN WITH "(" AND END WITH ")"',//,T20,'- EN
     >TER/RETURN TO CONTINUE -')
204   FORMAT(/,'FATAL ERROR IN SUBROUTINE DFCUS:',//,'THE NON-TRAILING-B
     >LANK LENGTH OF THE DEFAULT FORMAT CHARACTER',/,'BUFFER, LDFmt =',I
     >3,' IS GREATER THAN THE LENGTH OF THE FORMAT',/,'CHARACTER BUFFER,
     > LFmt =',I3,//,'EDIT AND RECOMPILE THE FORTRAN CODE.')
205   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE DFCUS')
      END
      SUBROUTINE NDRUS(NBuf,KUnit,*,*)
C
C
C***********************************************************************
C*                                                                     *
C*          NAMELIST-DIRECTED READ UTILITY SUBROUTINE (NDRUS)         *
C*                                                                     *
C*                    REVISION DATE: 23 MARCH 1999                     *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*  SUBROUTINE NDRUS EXECUTES AN INTELLIGENT  NAMELIST-DIRECTED READ   *
C*  PROCESS FOR  PROGRAM PSCCPP WITH INPUT VIA THE TEMPORARY SCRATCH   *
C*  FILE ON LOGICAL UNIT NUMBER Unit.                                  *
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* IBS    = INPUT CHARACTER BUFFER SIZE                                *
C*                                                                     *
C* VARIABLES:                                                          *
C*                                                                     *
C* NBuf   = NAMELIST GROUP-NAME CHARACTER BUFFER                       *
C* IBuf   = INPUT CHARACTER BUFFER                                     *
C* LUnit  = INPUT TEMPORARY SCRATCH FILE LOGICAL UNIT NUMBER           *
C*                                                                     *
C***********************************************************************
C
C
      CHARACTER AEnd*4,Amsand*1,Blank*1,Comma*1,DEnd*4,Dollar*1,IBuf*80,
     >NBuf*(*),RBuf*80,Switch*1
C
      INTEGER EPontr,SPontr,TPontr
C
      LOGICAL OP
C
      PARAMETER (IBS=80)
C
      EQUIVALENCE (Amsand,AEnd(1:1)),(Dollar,DEnd(1:1))
C
      SAVE AEnd,Blank,Comma,DEnd,EPontr,LUnit,SPontr,Switch,TPontr
C
      DATA AEnd/'&END'/,Blank/' '/,Comma/','/,DEnd/'$END'/,Switch/'U'/
C
C
C***********************************************************************
C*                                                                     *
C*       CHECK FOR A VALID NAMELIST GROUP-NAME CHARACTER BUFFER        *
C*                                                                     *
C***********************************************************************
C
C
      LUnit=KUnit
      CALL CBUS04(NBuf,Switch,*110)
      CALL CBUS03(NBuf)
      CALL CBUS01(NBuf,L)
      IF(L.EQ.0) GO TO 111
      IF(L+2.GT.IBS) GO TO 112
C
```

```
C
C***********************************************************************
C*                                                                     *
C*    OPEN THE TEMPORARY SCRATCH FILE ON LOGICAL UNIT NUMBER LUnit     *
C*                                                                     *
C***********************************************************************
C
C
      INQUIRE(UNIT=LUnit,ERR=113,OPENED=OP)
      IF(OP.EQV..TRUE.) CLOSE(UNIT=LUnit,STATUS='DELETE',ERR=114)
      OPEN(UNIT=LUnit,STATUS='SCRATCH',FORM='FORMATTED',ERR=115)
C
C
C***********************************************************************
C*                                                                     *
C*     SUPPLY THE NAMELIST INITIALIZATION AND NAMELIST GROUP-NAME      *
C*                                                                     *
C***********************************************************************
C
C
      CALL CBUS07(RBuf)
      RBuf(2:2)=Dollar
      RBuf(3:L+3)=NBuf(1:L)
      WRITE(UNIT=LUnit,FMT=201,ERR=116)RBuf
C
C
C***********************************************************************
C*                                                                     *
C*                   READ THE NAMELIST INPUT BUFFER                    *
C*                                                                     *
C***********************************************************************
C
C
101   READ(UNIT=5,FMT=201,ERR=101)IBuf
C
C
C***********************************************************************
C*                                                                     *
C*                   CHECK FOR A BLANK INPUT BUFFER                    *
C*                                                                     *
C***********************************************************************
C
C
      CALL CBUS01(IBuf,J)
      IF(J.NE.0) GO TO 102
C
C
C***********************************************************************
C*                                                                     *
C*                    SUPPLY THE DEFAULT CONDITION                     *
C*                                                                     *
C***********************************************************************
C
C
      CALL CBUS07(IBuf)
      IBuf(2:2)=Dollar
      WRITE(UNIT=LUnit,FMT=201,ERR=116)IBuf
      GO TO 109
C
C
```

```
C***********************************************************************
C*                                                                     *
C*       SEARCH FOR THE NAMELIST GROUP-NAME IN THE INPUT BUFFER        *
C*                                                                     *
C***********************************************************************
C
C
102   SPontr=1
      CALL CBUS04(IBuf,Switch,*110)
      CALL CBUS01(IBuf,EPontr)
      TPontr=INDEX(IBuf(SPontr:EPontr),NBuf(1:L))
      IF(TPontr.EQ.0) GO TO 108
C
C
C***********************************************************************
C*                                                                     *
C*                  CHECK FOR A VALID NAMELIST INPUT                   *
C*                                                                     *
C***********************************************************************
C
C
      SPontr=TPontr-1
      IF(SPontr.LT.1) GO TO 104
      DO 103 I=1,SPontr
      IF(IBuf(I:I).EQ.Blank) GO TO 103
      IF(IBuf(I:I).EQ.Dollar) GO TO 103
      IF(IBuf(I:I).EQ.Amsand) GO TO 103
      RETURN 1
103   CONTINUE
C
104   SPontr=TPontr+L
      IF(SPontr.GT.EPontr) GO TO 105
      IF(IBuf(SPontr:SPontr).EQ.Blank) GO TO 105
      IF(IBuf(SPontr:SPontr).EQ.Comma) GO TO 105
      IF(IBuf(SPontr:SPontr).EQ.Dollar) GO TO 105
      IF(IBuf(SPontr:SPontr).EQ.Amsand) GO TO 105
      RETURN 1
C
105   SPontr=SPontr-1
      DO 106 I=1,SPontr
106   IBuf(I:I)=Blank
      GO TO 108
C
C
C***********************************************************************
C*                                                                     *
C*             CONTINUE TO READ AND FLUSH THE INPUT BUFFER             *
C*                                                                     *
C***********************************************************************
C
C
107   READ(UNIT=5,FMT=201,ERR=107)IBuf
      CALL CBUS04(IBuf,Switch,*110)
      CALL CBUS01(IBuf,EPontr)
108   WRITE(UNIT=LUnit,FMT=201,ERR=116)IBuf
C
C
C***********************************************************************
C*                                                                     *
C*              CHECK FOR TERMINATION IN THE INPUT BUFFER              *
```

```
C*                                                                     *
C***********************************************************************
C
C
      SPontr=EPontr-3
      IF(IBuf(EPontr:EPontr).EQ.Dollar) GO TO 109
      IF(IBuf(EPontr:EPontr).EQ.Amsand) GO TO 109
      IF(IBuf(SPontr:EPontr).EQ.DEnd) GO TO 109
      IF(IBuf(SPontr:EPontr).EQ.AEnd) GO TO 109
      GO TO 107
109   REWIND LUnit
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*    CLOSE THE TEMPORARY SCRATCH FILE ON LOGICAL UNIT NUMBER UNIT     *
C*                                                                     *
C***********************************************************************
C
C
      ENTRY NDRUSE(*,*)
      CLOSE(UNIT=LUnit,STATUS='DELETE',ERR=114)
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
110   WRITE(UNIT=6,FMT=202)
      RETURN 2
111   WRITE(UNIT=6,FMT=203)
      RETURN 2
112   WRITE(UNIT=6,FMT=204)
      CALL CBUS06(6,NBuf,*117)
      WRITE(UNIT=6,FMT=205)IBS
      RETURN 2
113   WRITE(UNIT=6,FMT=206)LUnit
      RETURN 2
114   WRITE(UNIT=6,FMT=207)LUnit
      RETURN 2
115   WRITE(UNIT=6,FMT=208)LUnit
      RETURN 2
116   WRITE(UNIT=6,FMT=209)LUnit
      RETURN 2
117   WRITE(UNIT=6,FMT=210)
      RETURN 2
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
```

```
201   FORMAT(A80)
202   FORMAT('SUBROUTINE CBUS04 WAS CALLED FROM SUBROUTINE NDRUS')
203   FORMAT(/,'ERROR IN SUBROUTINE NDRUS: EMPTY NAMELIST GROUP-NAME CHA
     >RACTER',/,'BUFFER, NBuf')
204   FORMAT(/,'ERROR IN SUBROUTINE NDRUS: THE LENGTH OF THE NAMELIST GR
     >OUP-NAME NBuf =')
205   FORMAT('PLUS 2 EXCEEDS THE INPUT BUFFER SIZE, IBS =',I3)
206   FORMAT(/,'ERROR IN SUBROUTINE NDRUS: INQUIRE ERROR ON UNIT ',I1)
207   FORMAT(/,'ERROR IN SUBROUTINE NDRUS: CLOSE ERROR ON UNIT ',I1,/,'S
     >TATUS = "DELETE"')
208   FORMAT(/,'ERROR IN SUBROUTINE NDRUS: OPEN ERROR ON UNIT ',I1,', ST
     >ATUS = "SCRATCH"')
209   FORMAT(/,'ERROR IN SUBROUTINE NDRUS: WRITE ERROR ON UNIT ',I1)
210   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE NDRUS')
      END
      SUBROUTINE NDWUS(NBuf,KUnit,*)
C
C
C***********************************************************************
C*                                                                     *
C*         NAMELIST-DIRECTED WRITE UTILITY SUBROUTINE (NDWUS)          *
C*                                                                     *
C*                    REVISION DATE: 23 MARCH 1999                     *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C*  SUBROUTINE NDWUS EXECUTES AN INTELLIGENT NAMELIST-DIRECTED WRITE   *
C*  PROCESS FOR  PROGRAM PSCCPP  WITH OUTPUT TO THE  HIGH LEVEL PLOT   *
C*  COMMAND FILE ON LOGICAL UNIT NUMBER LUnit.
C*                                                                     *
C* PARAMETERS:                                                         *
C*                                                                     *
C* FieldL = FIELD LENGTH                                               *
C* OBS    = OUTPUT CHARACTER BUFFER SIZE                               *
C*                                                                     *
C* VARIABLES:                                                          *
C*                                                                     *
C* AFL    = AVAILABLE FIELD LENGTH                                     *
C* CBuf   = CHARACTER BUFFER                                           *
C* DBuf   = DOUBLE PRECISION ARRAY BUFFER                              *
C* IBuf   = INTEGER ARRAY BUFFER                                       *
C* N      = NUMBER OF ELEMENTS IN RBuf OR IBuf                         *
C* N1     = STARTING ELEMENT IN RBuf OR IBuf                           *
C* N2     = ENDING ELEMENT IN RBuf OR IBuf                             *
C* NBuf   = NAMELIST GROUP-NAME CHARACTER BUFFER                       *
C* NNFL   = NAMELIST NAME FIELD LENGTH                                 *
C* OBuf   = OUTPUT CHARACTER BUFFER                                    *
C* RBuf   = REAL ARRAY BUFFER                                          *
C* RFL    = REQUIRED FIELD LENGTH                                      *
C* LUnit  = OUTPUT FILE LOGICAL UNIT NUMBER                            *
C* VBuf   = VARIABLE NAME CHARACTER BUFFER                             *
C* VNFL   = VARIABLE NAME FIELD LENGTH                                 *
C*                                                                     *
C***********************************************************************
C
C
      CHARACTER Apos*2,CBuf*(*),Dollar*1,Equal*1,NBuf*(*),OBuf*75,
     >VBuf*(*)
C
      REAL*8 DBuf
C
      INTEGER AFL,EPontr,FieldL,OBS,RFL,SPontr,TPontr,VNFL
C
      PARAMETER (FieldL=15,OBS=75)
C
      DIMENSION DBuf(N),IBuf(N),RBuf(N)
C
      EQUIVALENCE (AFL,NNFL,VNFL)
C
      SAVE AFL,Apos,Dollar,EPontr,Equal,ICV,LUnit,OBuf,RFL,SPontr,
     >TPontr,VNFL
C
      DATA Apos/"',"/,Dollar/'$'/,Equal/'='/
```

```
C
C
C***********************************************************************
C*                                                                     *
C*                   OPEN THE NAMELIST OUTPUT BUFFER                    *
C*                                                                     *
C***********************************************************************
C
C
      LUnit=KUnit
      CALL CBUS03(NBuf)
      CALL CBUS01(NBuf,L)
      NNFL=FieldL
      IF((L+2).GT.NNFL) GO TO 124
      CALL CBUS07(OBuf)
      SPontr=2
      EPontr=2
      OBuf(SPontr:EPontr)=Dollar
      SPontr=EPontr+1
      EPontr=EPontr+L
      OBuf(SPontr:EPontr)=NBuf(1:L)
      TPontr=FieldL
      CALL CBUS01(OBuf,L)
      VNFL=TPontr-L-1
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*         INSERT A CHARACTER VARIABLE INTO THE OUTPUT BUFFER          *
C*                                                                     *
C***********************************************************************
C
C
      ENTRY NDWUSC(VBuf,CBuf,*)
      CALL CBUS03(CBuf)
      CALL CBUS01(CBuf,L)
      RFL=L+4
      ICV=1
      GO TO 118
101   CALL CBUS01(CBuf,L)
      AFL=0
102   TPontr=TPontr+FieldL
      AFL=AFL+FieldL
      IF(RFL.GT.AFL) GO TO 102
      IF(TPontr.GT.OBS) TPontr=OBS
      SPontr=TPontr-L-2
      EPontr=SPontr
      OBuf(SPontr:EPontr)=Apos(1:1)
      SPontr=EPontr+1
      EPontr=TPontr-2
      OBuf(SPontr:EPontr)=CBuf(1:L)
      SPontr=TPontr-1
      EPontr=TPontr
      OBuf(SPontr:EPontr)=Apos
      VNFL=0
      RETURN
C
C
C***********************************************************************
```

```
C*                                                                     *
C*          INSERT AN INTEGER ARRAY INTO THE OUTPUT BUFFER            *
C*                                                                     *
C***********************************************************************
C
C
      ENTRY NDWUSI(VBuf,N,N1,N2,IBuf,*)
      RFL=FieldL
      ICV=2
      GO TO 118
103   DO 107 I=N1,N2
104   TPontr=TPontr+FieldL
      IF(TPontr.LE.OBS) GO TO 106
      JCV=1
      GO TO 123
105   GO TO 104
106   SPontr=TPontr-FieldL+1
      EPontr=TPontr
107   WRITE(OBuf(SPontr:EPontr),FMT=201)IBuf(I)
      VNFL=0
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*            INSERT A REAL ARRAY INTO THE OUTPUT BUFFER              *
C*                                                                     *
C***********************************************************************
C
C
      ENTRY NDWUSR(VBuf,N,N1,N2,RBuf,*)
      RFL=FieldL
      ICV=3
      GO TO 118
108   DO 112 I=N1,N2
109   TPontr=TPontr+FieldL
      IF(TPontr.LE.OBS) GO TO 111
      JVC=2
      GO TO 123
110   GO TO 109
111   SPontr=TPontr-FieldL+1
      EPontr=TPontr
112   WRITE(OBuf(SPontr:EPontr),FMT=202)RBuf(I)
      VNFL=0
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*      INSERT A DOUBLE PRECISION ARRAY INTO THE OUTPUT BUFFER        *
C*                                                                     *
C***********************************************************************
C
C
      ENTRY NDWUSD(VBuf,N,N1,N2,DBuf,*)
      RFL=FieldL
      ICV=4
      GO TO 118
113   DO 117 I=N1,N2
114   TPontr=TPontr+FieldL
```

```
      IF(TPontr.LE.OBS) GO TO 116
      JVC=3
      GO TO 123
115   GO TO 114
116   SPontr=TPontr-FieldL+1
      EPontr=TPontr
117   WRITE(OBuf(SPontr:EPontr),FMT=202)DBuf(I)
      VNFL=0
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*            INSERT A VARIABLE NAME INTO THE OUTPUT BUFFER            *
C*                                                                     *
C***********************************************************************
C
C
118   CALL CBUS03(VBuf)
      CALL CBUS01(VBuf,L)
      IF((L+2)+RFL.GT.OBS) RETURN 1
119   IF((L+2).LE.VNFL) GO TO 120
      TPontr=TPontr+FieldL
      VNFL=VNFL+FieldL
      GO TO 119
120   IF(TPontr+RFL.LE.OBS) GO TO 122
      JVC=4
      GO TO 123
121   GO TO 119
122   SPontr=TPontr-L-1
      EPontr=TPontr-2
      OBuf(SPontr:EPontr)=VBuf(1:L)
      SPontr=EPontr+2
      EPontr=TPontr
      OBuf(SPontr:EPontr)=Equal
      GO TO (101,103,108,113),ICV
C
C
C***********************************************************************
C*                                                                     *
C*                       FLUSH THE OUTPUT BUFFER                       *
C*                                                                     *
C***********************************************************************
C
C
123   WRITE(UNIT=LUnit,Fmt=203,ERR=129)OBuf
      CALL CBUS07(OBuf)
      TPontr=0
      VNFL=0
      GO TO (105,110,115,121),JVC
C
C
C***********************************************************************
C*                                                                     *
C*                  CLOSE THE NAMELIST OUTPUT BUFFER                   *
C*                                                                     *
C***********************************************************************
C
C
      ENTRY NDWUSE(*)
      CALL CBUS01(OBuf,L)
      SPontr=L
      EPontr=L
      OBuf(SPontr:EPontr)=Dollar
      WRITE(UNIT=LUnit,Fmt=203,ERR=129)OBuf
      IF(LUnit.EQ.6) WRITE(UNIT=LUnit,Fmt=204,ERR=129)
      RETURN
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
124   WRITE(UNIT=6,FMT=205)
      CALL CBUS06(6,NBuf,*130)
      WRITE(UNIT=6,FMT=206)NNFL
      RETURN 1
125   WRITE(UNIT=6,FMT=207)
      CALL CBUS06(6,VBuf,*130)
      WRITE(UNIT=6,FMT=208)
      CALL CBUS06(6,CBuf,*130)
      WRITE(UNIT=6,FMT=209)OBS
      RETURN 1
126   WRITE(UNIT=6,FMT=207)
      CALL CBUS06(6,VBuf,*130)
      WRITE(UNIT=6,FMT=210)OBS
      RETURN 1
127   WRITE(UNIT=6,FMT=207)
      CALL CBUS06(6,VBuf,*130)
      WRITE(UNIT=6,FMT=211)OBS
      RETURN 1
128   WRITE(UNIT=6,FMT=207)
      CALL CBUS06(6,VBuf,*130)
      WRITE(UNIT=6,FMT=212)OBS
      RETURN 1
129   WRITE(UNIT=6,FMT=213)LUnit
      RETURN 1
130   WRITE(UNIT=6,FMT=214)
131   RETURN 1
C
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(I14,',')
202   FORMAT(E14.6,',')
203   FORMAT(A75)
204   FORMAT()
205   FORMAT(/,'ERROR IN SUBROUTINE NDWUS: THE LENGTH OF THE NAMELIST GR
     >OUP-NAME NBuf =')
206   FORMAT('EXCEEDS THE NAMELIST NAME FIELD LENGTH =',I3)
207   FORMAT(/,'ERROR IN SUBROUTINE NDWUS: THE LENGTH OF THE VARIABLE NA
     >ME VBuf =')
```

```
208   FORMAT('PLUS THE LENGTH OF THE CHARACTER VARIABLE CBuf =')
209   FORMAT('EXCEEDS THE OUTPUT BUFFER SIZE, OBS =',I3)
210   FORMAT('PLUS THE FIELD LENGTH FOR AN INTEGER VARIABLE EXCEEDS THE
     >OUTPUT BUFFER',/,'SIZE, OBS =',I3)
211   FORMAT('PLUS THE FIELD LENGTH FOR A REAL VARIABLE EXCEEDS THE OUTP
     >UT BUFFER',/,'SIZE, OBS =',I3)
212   FORMAT('PLUS THE FIELD LENGTH FOR A DOUBLE PRECISION VARIABLE EXCE
     >EDS THE OUTPUT BUFFER',/,'SIZE, OBS =',I3)
213   FORMAT(/,'ERROR IN SUBROUTINE NDWUS: WRITE ERROR ON UNIT ',I1)
214   FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE NDWUS')
      END
      SUBROUTINE YNOUS(*,*,*)
C
C
C***********************************************************************
C*                                                                     *
C*               YES/NO/OTHER UTILITY SUBROUTINE (YNOUS)               *
C*                                                                     *
C*                  REVISION DATE: 29 SEPTEMBER 2000                   *
C*                                                                     *
C***********************************************************************
C
C***********************************************************************
C*                                                                     *
C* SUBROUTINE YNOUS READS ONE BYTE FROM UNIT 5  ANTICIPATING A 'Y' OR  *
C* 'y' YES RESPONSE OR ALTERNATIVELY A 'N' OR 'n' NO RESPONSE.  A YES  *
C* RESPONSE  RESULTS  IN A  RETURN 1;  A NO  RESPONSE  RESULTS  IN  A  *
C* RETURN 2.  ALL OTHER RESPONSES RESULT IN A RETURN 3.                *
C*                                                                     *
C***********************************************************************
C
C
      CHARACTER CCV*1,No*1,Switch*1,Yes*1
C
      SAVE No,Yes
C
      DATA No/'N'/,Switch/'U'/,Yes/'Y'/
C
C
C***********************************************************************
C*                                                                     *
C*                READ ONE BYTE AND COMPARE WITH Yes/No                *
C*                                                                     *
C***********************************************************************
C
C
      READ(UNIT=5,FMT=201)CCV
      CALL CBUS04(CCV,Switch,*101)
      IF(CCV.EQ.Yes) RETURN 1
      IF(CCV.EQ.No) RETURN 2
      RETURN 3
C
C
C***********************************************************************
C*                                                                     *
C*                           ERROR MESSAGES                            *
C*                                                                     *
C***********************************************************************
C
C
101   WRITE(UNIT=6,FMT=202)
      RETURN 3
C
C***********************************************************************
C*                                                                     *
C*                          FORMAT STATEMENTS                          *
C*                                                                     *
C***********************************************************************
C
C
201   FORMAT(A1)
```

```
202   FORMAT('SUBROUTINE CBUS04 WAS CALLED FROM SUBROUTINE YNOUS')
      END
```

APPENDIX D
EXAMPLE 1

In this example, the LSPRWC code was applied to a set of observations taken for an object accelerating from rest at a constant 9.81 m/s^2, the acceleration of gravity. The true behavior for the object, in terms of displacement as a function of time, is given by

$$\frac{d^2x}{dt^2} = \text{constant} \tag{1}$$

$$= a, \tag{2}$$

and integrating equation (1) twice gives

$$\frac{dx}{dt} = v_0 + at, \tag{3}$$

$$= v(t), \tag{4}$$

and

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2 \tag{5}$$

where

$a =$ acceleration,
$t =$ time,
$v =$ velocity,
$v_0 =$ initial velocity,
$x =$ displacement,
$x_0 =$ initial position.

Then for

$$x_0 = 0, \tag{6}$$

$$v_0 = 0, \tag{7}$$

equations (3), (4), and (5) reduce to

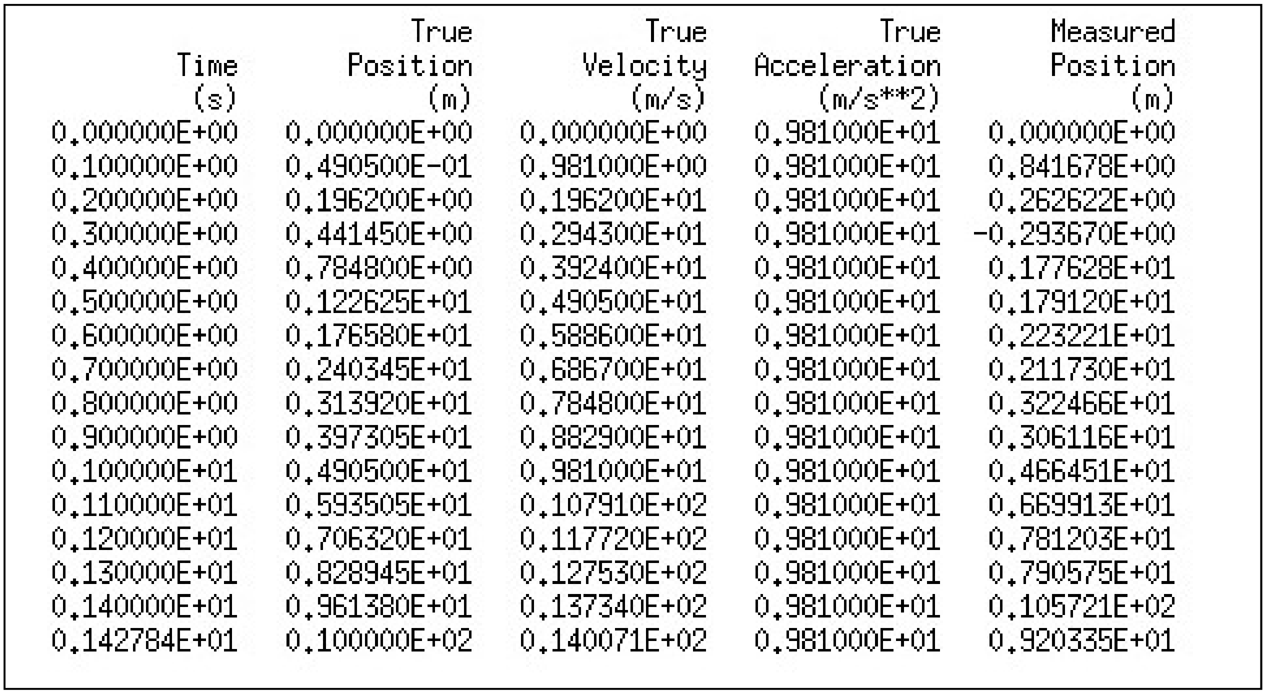
$$v(t) = at \tag{8}$$

and

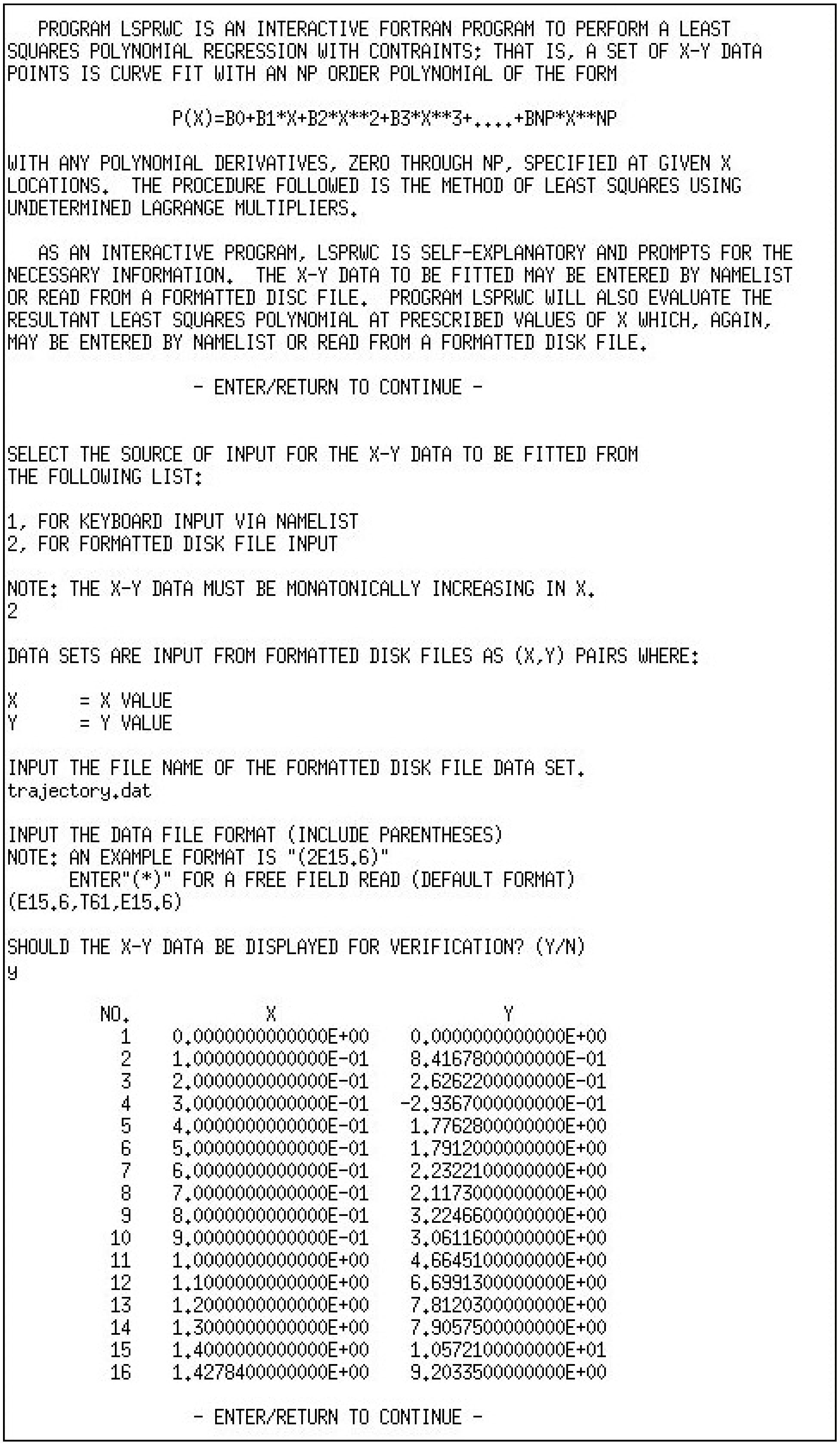
$$x(t) = \frac{1}{2} a t^2. \tag{9}$$

The observations for this example are given in Table D-1 along with the true values for displacement, velocity, and acceleration. Figures D-1 through 3 present *X Window System* screen shots from LSPRWC code execution using this data set.

Table D-1. Trajectory Data

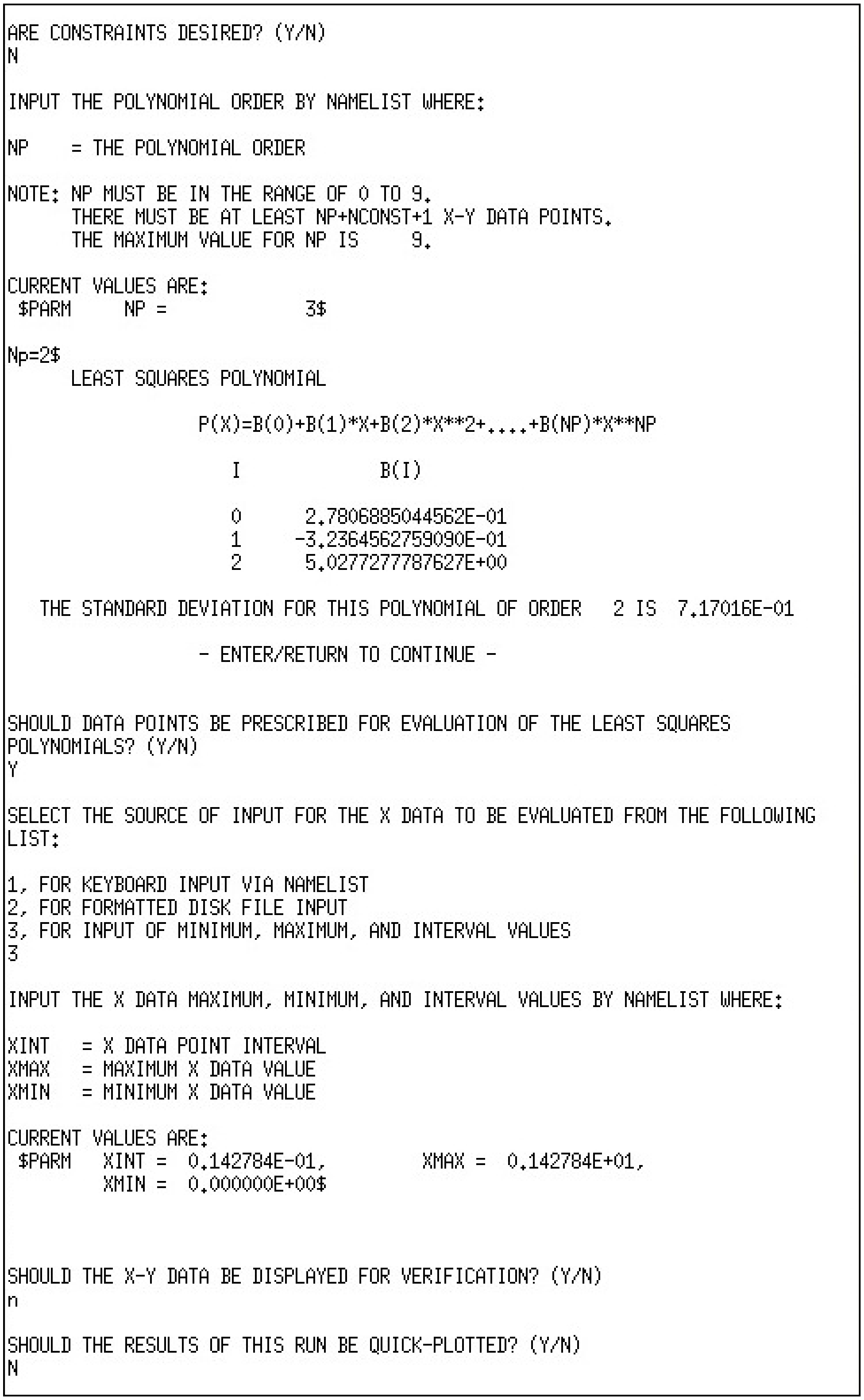


Time (s)	True Position (m)	True Velocity (m/s)	True Acceleration (m/s**2)	Measured Position (m)
0.000000E+00	0.000000E+00	0.000000E+00	0.981000E+01	0.000000E+00
0.100000E+00	0.490500E-01	0.981000E+00	0.981000E+01	0.841678E+00
0.200000E+00	0.196200E+00	0.196200E+01	0.981000E+01	0.262622E+00
0.300000E+00	0.441450E+00	0.294300E+01	0.981000E+01	-0.293670E+00
0.400000E+00	0.784800E+00	0.392400E+01	0.981000E+01	0.177628E+01
0.500000E+00	0.122625E+01	0.490500E+01	0.981000E+01	0.179120E+01
0.600000E+00	0.176580E+01	0.588600E+01	0.981000E+01	0.223221E+01
0.700000E+00	0.240345E+01	0.686700E+01	0.981000E+01	0.211730E+01
0.800000E+00	0.313920E+01	0.784800E+01	0.981000E+01	0.322466E+01
0.900000E+00	0.397305E+01	0.882900E+01	0.981000E+01	0.306116E+01
0.100000E+01	0.490500E+01	0.981000E+01	0.981000E+01	0.466451E+01
0.110000E+01	0.593505E+01	0.107910E+02	0.981000E+01	0.669913E+01
0.120000E+01	0.706320E+01	0.117720E+02	0.981000E+01	0.781203E+01
0.130000E+01	0.828945E+01	0.127530E+02	0.981000E+01	0.790575E+01
0.140000E+01	0.961380E+01	0.137340E+02	0.981000E+01	0.105721E+02
0.142784E+01	0.100000E+02	0.140071E+02	0.981000E+01	0.920335E+01



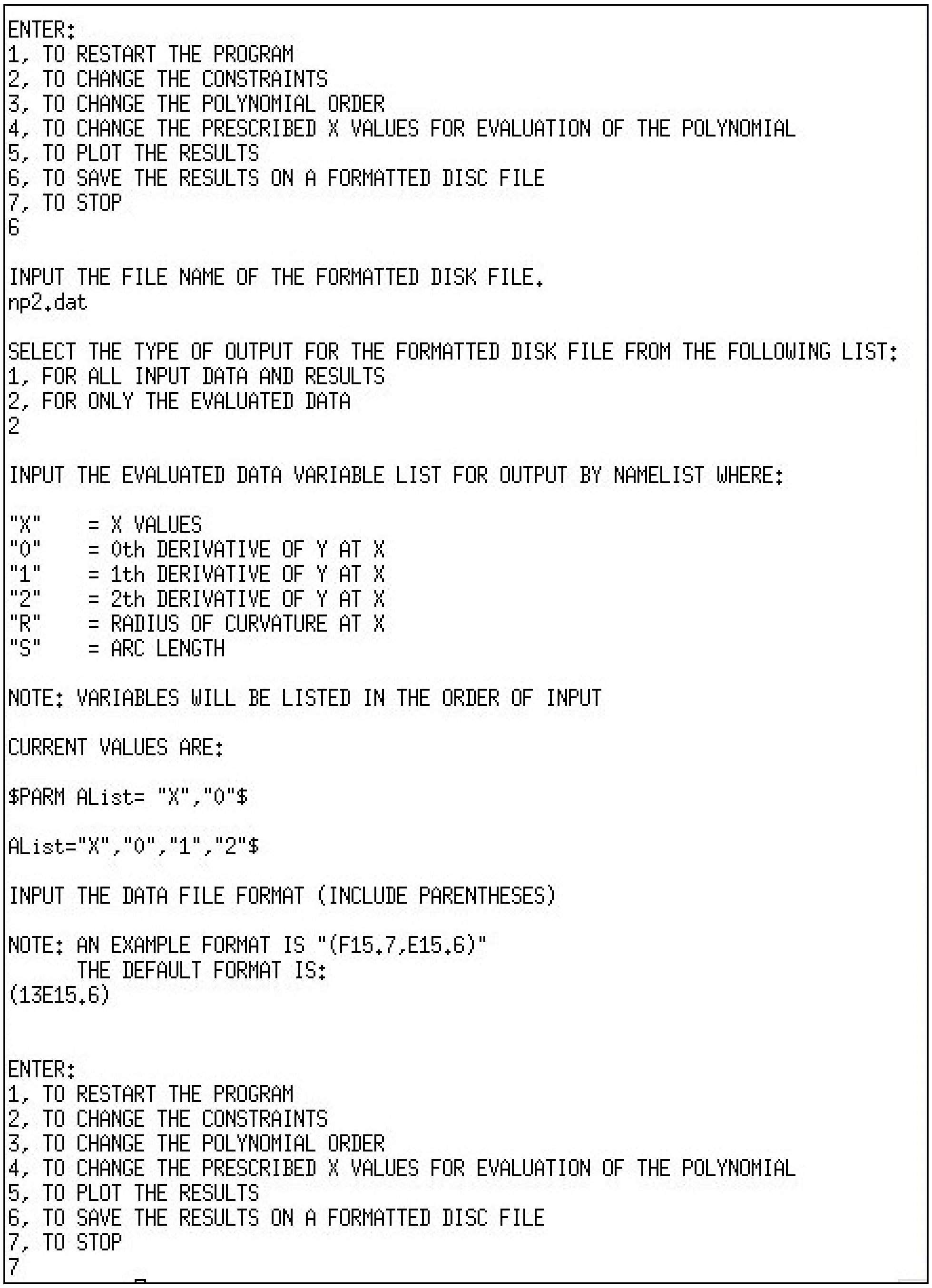
```
   PROGRAM LSPRWC IS AN INTERACTIVE FORTRAN PROGRAM TO PERFORM A LEAST
SQUARES POLYNOMIAL REGRESSION WITH CONTRAINTS; THAT IS, A SET OF X-Y DATA
POINTS IS CURVE FIT WITH AN NP ORDER POLYNOMIAL OF THE FORM

                P(X)=B0+B1*X+B2*X**2+B3*X**3+....+BNP*X**NP

WITH ANY POLYNOMIAL DERIVATIVES, ZERO THROUGH NP, SPECIFIED AT GIVEN X
LOCATIONS.  THE PROCEDURE FOLLOWED IS THE METHOD OF LEAST SQUARES USING
UNDETERMINED LAGRANGE MULTIPLIERS.

   AS AN INTERACTIVE PROGRAM, LSPRWC IS SELF-EXPLANATORY AND PROMPTS FOR THE
NECESSARY INFORMATION.  THE X-Y DATA TO BE FITTED MAY BE ENTERED BY NAMELIST
OR READ FROM A FORMATTED DISC FILE.  PROGRAM LSPRWC WILL ALSO EVALUATE THE
RESULTANT LEAST SQUARES POLYNOMIAL AT PRESCRIBED VALUES OF X WHICH, AGAIN,
MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE.

                  - ENTER/RETURN TO CONTINUE -


SELECT THE SOURCE OF INPUT FOR THE X-Y DATA TO BE FITTED FROM
THE FOLLOWING LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT

NOTE: THE X-Y DATA MUST BE MONATONICALLY INCREASING IN X.
2

DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS (X,Y) PAIRS WHERE:

X      = X VALUE
Y      = Y VALUE

INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
trajectory.dat

INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)
NOTE: AN EXAMPLE FORMAT IS "(2E15.6)"
      ENTER"(*)" FOR A FREE FIELD READ (DEFAULT FORMAT)
(E15.6,T61,E15.6)

SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
y

         NO.             X                      Y
           1    0.0000000000000E+00    0.0000000000000E+00
           2    1.0000000000000E-01    8.4167800000000E-01
           3    2.0000000000000E-01    2.6262200000000E-01
           4    3.0000000000000E-01   -2.9367000000000E-01
           5    4.0000000000000E-01    1.7762800000000E+00
           6    5.0000000000000E-01    1.7912000000000E+00
           7    6.0000000000000E-01    2.2322100000000E+00
           8    7.0000000000000E-01    2.1173000000000E+00
           9    8.0000000000000E-01    3.2246600000000E+00
          10    9.0000000000000E-01    3.0611600000000E+00
          11    1.0000000000000E+00    4.6645100000000E+00
          12    1.1000000000000E+00    6.6991300000000E+00
          13    1.2000000000000E+00    7.8120300000000E+00
          14    1.3000000000000E+00    7.9057500000000E+00
          15    1.4000000000000E+00    1.0572100000000E+01
          16    1.4278400000000E+00    9.2033500000000E+00

                  - ENTER/RETURN TO CONTINUE -
```

Figure D-1 LSPRWC *Screen Shot—Part 1*



```
ARE CONSTRAINTS DESIRED? (Y/N)
N

INPUT THE POLYNOMIAL ORDER BY NAMELIST WHERE:

NP    = THE POLYNOMIAL ORDER

NOTE: NP MUST BE IN THE RANGE OF 0 TO 9.
      THERE MUST BE AT LEAST NP+NCONST+1 X-Y DATA POINTS.
      THE MAXIMUM VALUE FOR NP IS     9.

CURRENT VALUES ARE:
 $PARM     NP =             3$

Np=2$
      LEAST SQUARES POLYNOMIAL

                  P(X)=B(0)+B(1)*X+B(2)*X**2+....+B(NP)*X**NP

                     I             B(I)

                     0      2.7806885044562E-01
                     1     -3.2364562759090E-01
                     2      5.0277277787627E+00

   THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER   2 IS  7.17016E-01

                  - ENTER/RETURN TO CONTINUE -


SHOULD DATA POINTS BE PRESCRIBED FOR EVALUATION OF THE LEAST SQUARES
POLYNOMIALS? (Y/N)
Y

SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATED FROM THE FOLLOWING
LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT
3, FOR INPUT OF MINIMUM, MAXIMUM, AND INTERVAL VALUES
3

INPUT THE X DATA MAXIMUM, MINIMUM, AND INTERVAL VALUES BY NAMELIST WHERE:

XINT   = X DATA POINT INTERVAL
XMAX   = MAXIMUM X DATA VALUE
XMIN   = MINIMUM X DATA VALUE

CURRENT VALUES ARE:
 $PARM   XINT =  0.142784E-01,         XMAX =  0.142784E+01,
         XMIN =  0.000000E+00$



SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
n

SHOULD THE RESULTS OF THIS RUN BE QUICK-PLOTTED? (Y/N)
N
```

Figure D-2. LSPRWC Screen Shot—Part 2



```
ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
6

INPUT THE FILE NAME OF THE FORMATTED DISK FILE.
np2.dat

SELECT THE TYPE OF OUTPUT FOR THE FORMATTED DISK FILE FROM THE FOLLOWING LIST:
1, FOR ALL INPUT DATA AND RESULTS
2, FOR ONLY THE EVALUATED DATA
2

INPUT THE EVALUATED DATA VARIABLE LIST FOR OUTPUT BY NAMELIST WHERE:

"X"    = X VALUES
"0"    = 0th DERIVATIVE OF Y AT X
"1"    = 1th DERIVATIVE OF Y AT X
"2"    = 2th DERIVATIVE OF Y AT X
"R"    = RADIUS OF CURVATURE AT X
"S"    = ARC LENGTH

NOTE: VARIABLES WILL BE LISTED IN THE ORDER OF INPUT

CURRENT VALUES ARE:

$PARM AList= "X","0"$

AList="X","0","1","2"$

INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)

NOTE: AN EXAMPLE FORMAT IS "(F15.7,E15.6)"
      THE DEFAULT FORMAT IS:
(13E15.6)


ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
7
```

Figure D-3. LSPRWC *Screen Shot—Part 3*

Following along through Figure D-1, the user was presented with an introduction to LSPRWC and then asked for the data source. Here the source was selected as a formatted disk file followed by the file name and read format. The (x, y) data pairs—time and measured position—were displayed for verification and match the values given in Table D-1.

Moving on through Figure D-2, the code asked if problem constraints were desired and none were selected. The user was then asked to enter the polynomial order which was set to $n_p = 2$ by means of NAMELIST. With the problem thus fully defined, program LSPRWC returned values for the coefficients of the least squares regression polynomial, b_0, b_1, and b_2, and the standard deviation for the fit to the (x, y) data pairs as input. This 2nd order regression polynomial was then evaluated at specific values of x, in this case time, using the minimum, maximum, and interval option.

Following through Figure D-3, menu options were chosen to output the evaluated data to a formatted disk file writing, in order, each value of x, y, and the first and second derivatives, $\frac{dy}{dx}$, and $\frac{d^2y}{dx^2}$. With that, the LSPRWC code menu option was selected for program termination.

The results from this particular LSPRWC run are shown in Figure D-4 as plots of displacement and velocity as a function of time along with the true values and the measurements of Table D-1. The 2nd order least squares regression polynomial fits quite well in this case, as might be expected considering that the true solution for displacement, given by equation (9), is 2nd order in time.

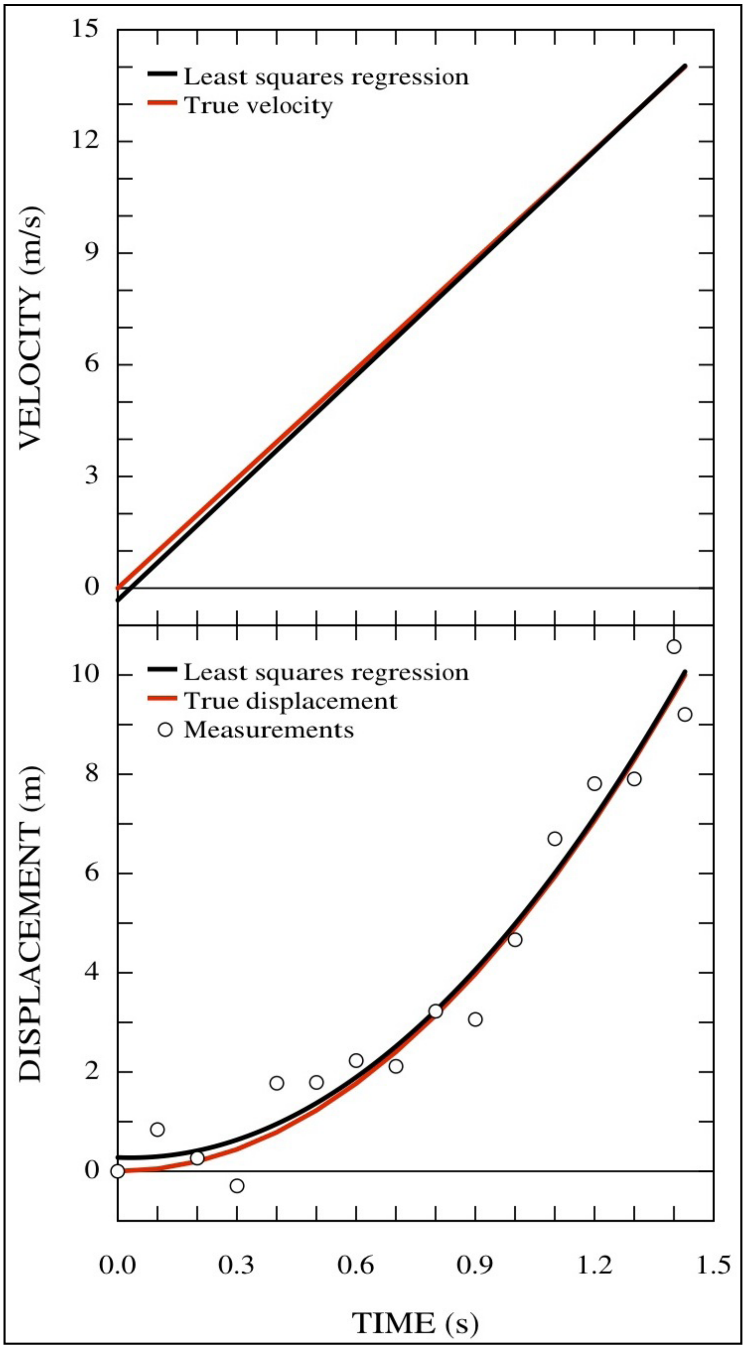


Figure D-4. Least Squares Polynomial Regression for n_p = 2

The LSPRWC code could have been run with any choice of polynomial order from 0 through 9 as was done to produce the results of Figure D-5. This plot of displacement as a function of time shows a reasonable fit for the regression polynomials for n_p = 2 and 3. Clearly, the representation begins to degrade with polynomial orders of 4 and greater.

Section V. LIMITATIONS notes that the LSPRWC code limits the polynomial order such that $0 \leq n_p \leq 9$. This example serves as excellent justification for this limitation since the

9th order least squares regression polynomial gives the best fit to the measured data in Figure D-5 yet the polynomial representation is quite unlike the true solution.

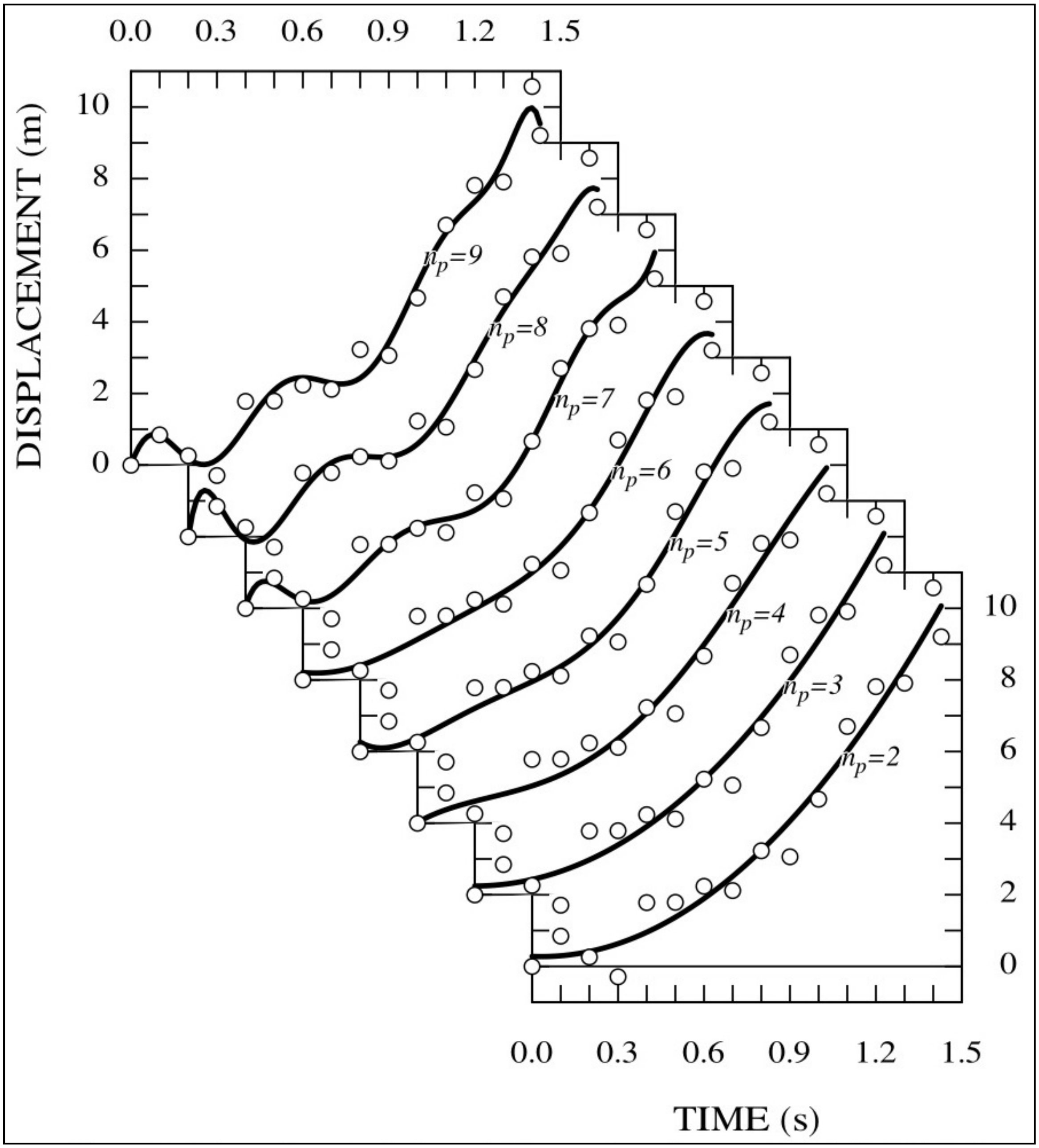
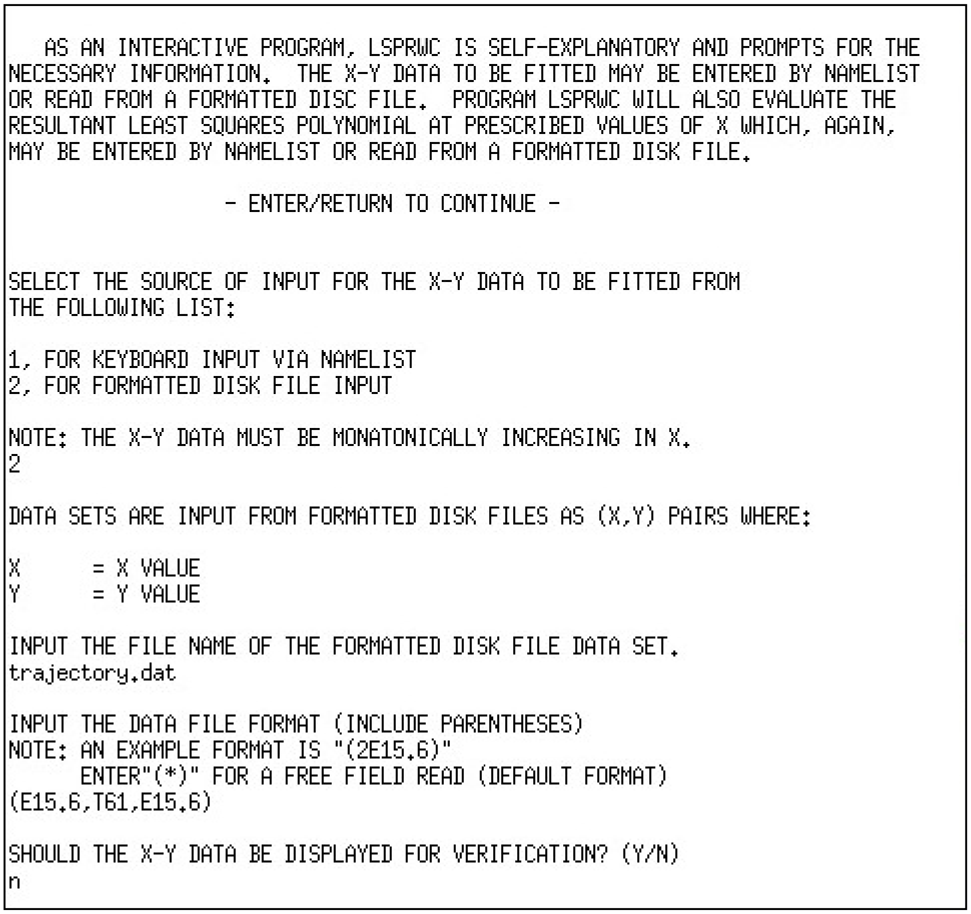


Figure D-5. Least Squares Polynomial Regressions for n_p = 2, 3, 4, 5, 6, 7, 8, 9

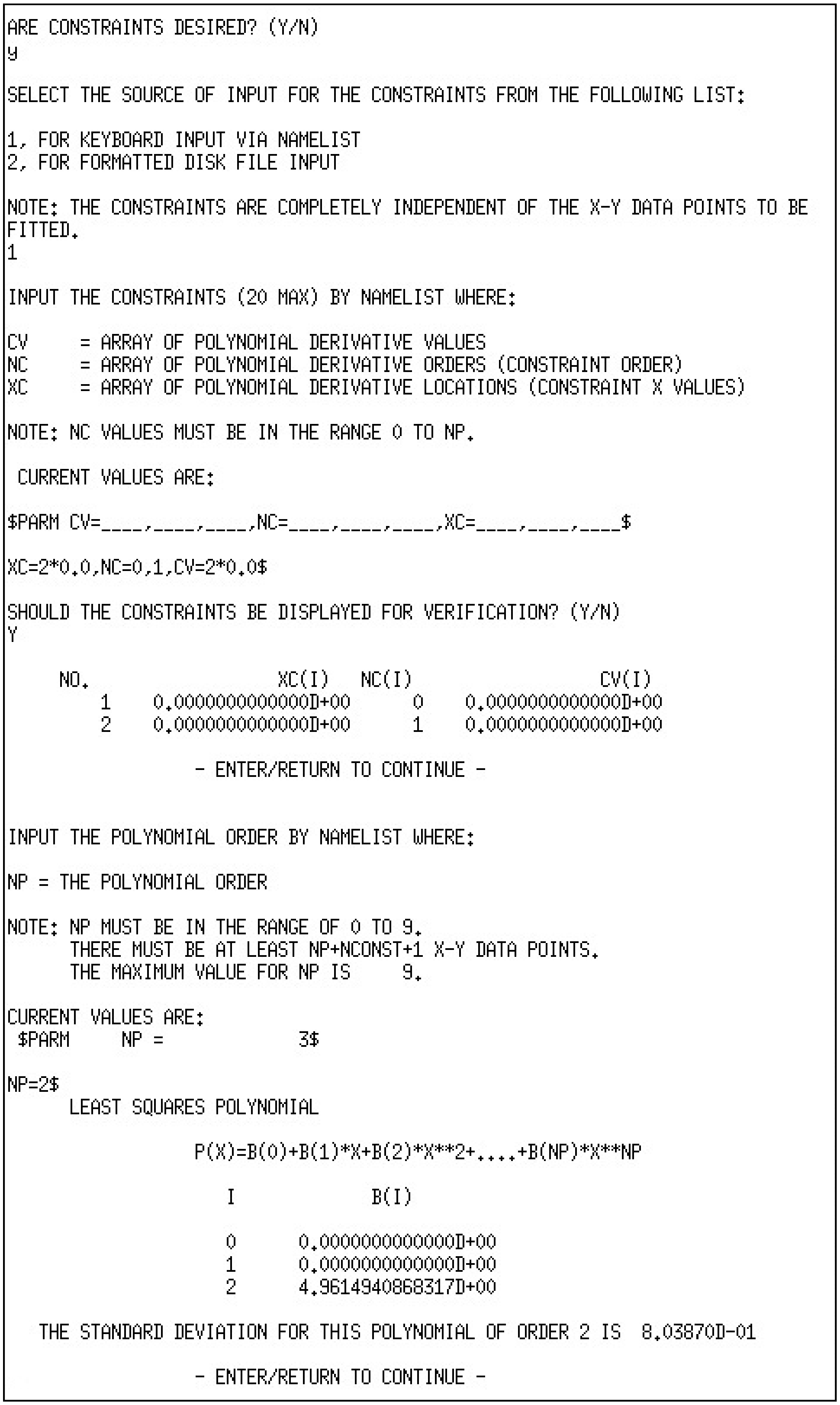
Examining once again the plots of displacement and velocity as a function of time in Figure D-4, it is obvious that this 2nd order regression polynomial does not satisfy either equation (6) or (7); that is to say, the displacement and velocity are non-zero at time equal zero. Furthermore, the polynomial coefficients b_0 and b_1 are non-zero as shown in Figure D-2. Still, the acceleration $a = 2 \times b_2 = 10.055$ m/s^2 is reasonably close to the true value of 9.81 m/s^2.

Since the initial displacement and velocity are known, these could be used as constraints with the LSPRWC code to improve the regression. Figures D-6 through 9 present *X Window System* screen shots from LSPRWC code execution using again the observations of Table D-1 but with these constraints.



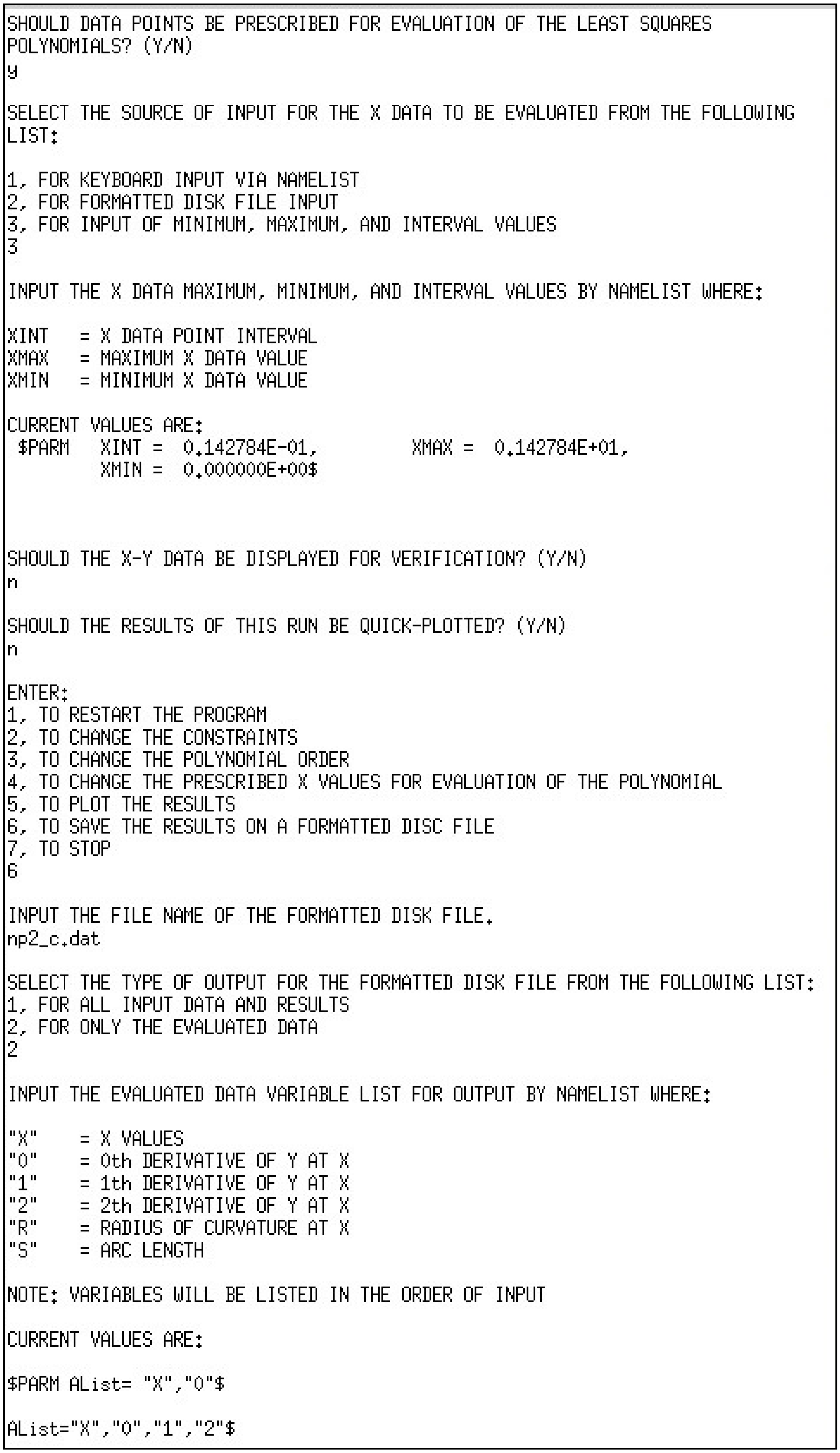
```
   AS AN INTERACTIVE PROGRAM, LSPRWC IS SELF-EXPLANATORY AND PROMPTS FOR THE
NECESSARY INFORMATION.  THE X-Y DATA TO BE FITTED MAY BE ENTERED BY NAMELIST
OR READ FROM A FORMATTED DISC FILE.  PROGRAM LSPRWC WILL ALSO EVALUATE THE
RESULTANT LEAST SQUARES POLYNOMIAL AT PRESCRIBED VALUES OF X WHICH, AGAIN,
MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE.

                  - ENTER/RETURN TO CONTINUE -


SELECT THE SOURCE OF INPUT FOR THE X-Y DATA TO BE FITTED FROM
THE FOLLOWING LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT

NOTE: THE X-Y DATA MUST BE MONATONICALLY INCREASING IN X.
2

DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS (X,Y) PAIRS WHERE:

X      = X VALUE
Y      = Y VALUE

INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
trajectory.dat

INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)
NOTE: AN EXAMPLE FORMAT IS "(2E15.6)"
      ENTER"(*)" FOR A FREE FIELD READ (DEFAULT FORMAT)
(E15.6,T61,E15.6)

SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
n
```

Figure D-6. LSPRWC Screen Shot—Part 1



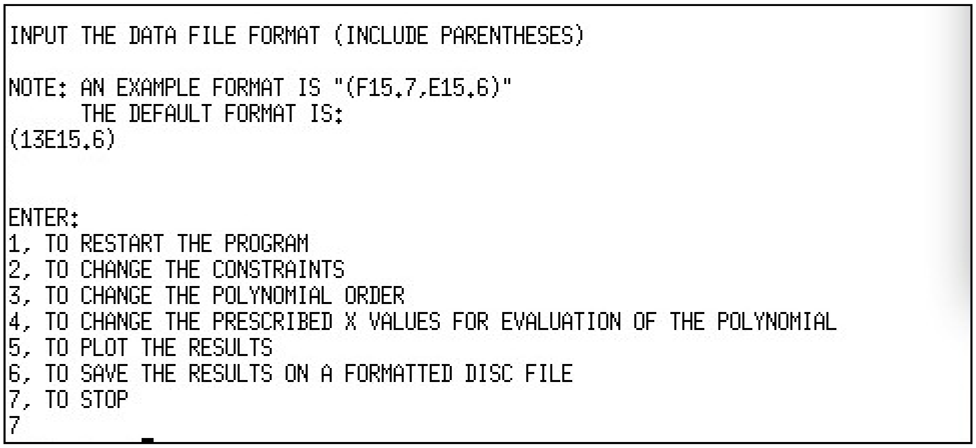
```
ARE CONSTRAINTS DESIRED? (Y/N)
y

SELECT THE SOURCE OF INPUT FOR THE CONSTRAINTS FROM THE FOLLOWING LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT

NOTE: THE CONSTRAINTS ARE COMPLETELY INDEPENDENT OF THE X-Y DATA POINTS TO BE
FITTED.
1

INPUT THE CONSTRAINTS (20 MAX) BY NAMELIST WHERE:

CV     = ARRAY OF POLYNOMIAL DERIVATIVE VALUES
NC     = ARRAY OF POLYNOMIAL DERIVATIVE ORDERS (CONSTRAINT ORDER)
XC     = ARRAY OF POLYNOMIAL DERIVATIVE LOCATIONS (CONSTRAINT X VALUES)

NOTE: NC VALUES MUST BE IN THE RANGE 0 TO NP.

 CURRENT VALUES ARE:

$PARM CV=____,____,____,NC=____,____,____,XC=____,____,____$

XC=2*0.0,NC=0,1,CV=2*0.0$

SHOULD THE CONSTRAINTS BE DISPLAYED FOR VERIFICATION? (Y/N)
Y

     NO.                 XC(I)   NC(I)                  CV(I)
         1    0.0000000000000D+00      0    0.0000000000000D+00
         2    0.0000000000000D+00      1    0.0000000000000D+00

                  - ENTER/RETURN TO CONTINUE -


INPUT THE POLYNOMIAL ORDER BY NAMELIST WHERE:

NP = THE POLYNOMIAL ORDER

NOTE: NP MUST BE IN THE RANGE OF 0 TO 9.
      THERE MUST BE AT LEAST NP+NCONST+1 X-Y DATA POINTS.
      THE MAXIMUM VALUE FOR NP IS     9.

CURRENT VALUES ARE:
 $PARM     NP =             3$

NP=2$
      LEAST SQUARES POLYNOMIAL

                  P(X)=B(0)+B(1)*X+B(2)*X**2+....+B(NP)*X**NP

                     I             B(I)

                     0      0.0000000000000D+00
                     1      0.0000000000000D+00
                     2      4.9614940868317D+00

   THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER 2 IS  8.03870D-01

                  - ENTER/RETURN TO CONTINUE -
```

Figure D-7. LSPRWC Screen Shot—Part 2



```
SHOULD DATA POINTS BE PRESCRIBED FOR EVALUATION OF THE LEAST SQUARES
POLYNOMIALS? (Y/N)
y

SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATED FROM THE FOLLOWING
LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT
3, FOR INPUT OF MINIMUM, MAXIMUM, AND INTERVAL VALUES
3

INPUT THE X DATA MAXIMUM, MINIMUM, AND INTERVAL VALUES BY NAMELIST WHERE:

XINT   = X DATA POINT INTERVAL
XMAX   = MAXIMUM X DATA VALUE
XMIN   = MINIMUM X DATA VALUE

CURRENT VALUES ARE:
 $PARM   XINT =  0.142784E-01,         XMAX =  0.142784E+01,
         XMIN =  0.000000E+00$



SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
n

SHOULD THE RESULTS OF THIS RUN BE QUICK-PLOTTED? (Y/N)
n

ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
6

INPUT THE FILE NAME OF THE FORMATTED DISK FILE.
np2_c.dat

SELECT THE TYPE OF OUTPUT FOR THE FORMATTED DISK FILE FROM THE FOLLOWING LIST:
1, FOR ALL INPUT DATA AND RESULTS
2, FOR ONLY THE EVALUATED DATA
2

INPUT THE EVALUATED DATA VARIABLE LIST FOR OUTPUT BY NAMELIST WHERE:

"X"    = X VALUES
"0"    = 0th DERIVATIVE OF Y AT X
"1"    = 1th DERIVATIVE OF Y AT X
"2"    = 2th DERIVATIVE OF Y AT X
"R"    = RADIUS OF CURVATURE AT X
"S"    = ARC LENGTH

NOTE: VARIABLES WILL BE LISTED IN THE ORDER OF INPUT

CURRENT VALUES ARE:

$PARM AList= "X","0"$

AList="X","0","1","2"$
```

Figure D-8. LSPRWC Screen Shot—Part 3



```
INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)

NOTE: AN EXAMPLE FORMAT IS "(F15.7,E15.6)"
      THE DEFAULT FORMAT IS:
(13E15.6)


ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
7
```

Figure D-9. LSPRWC *Screen Shot—Part 4*

Following along through Figure D-6, the user was presented with the introduction to LSPRWC, the data source was selected as a formatted disk file, and the file name and read format were entered.

Continuing through Figure D-7, the code asked if problem constraints were desired and a positive response was entered with the constraints defined by means of NAMELIST. The constraints were then displayed for verification to confirm that the 0^{th} and 1^{st} derivatives were set to zero at time zero. The polynomial order was again set to $n_p = 2$ and with the problem thus fully defined, program LSPRWC returned values for the coefficients of the least squares regression polynomial.

Following now through Figure D-8, this second-order regression polynomial was evaluated at specific values of *x* again using the minimum, maximum, and interval option. Menu options were then chosen to output the evaluated data to a formatted disk file writing, in order, each value of *x*, *y*, and the 1^{st} and 2^{nd} derivatives.

Finally, in Figure 9, the LSPRWC code menu option was selected for program termination.

The results from this LSPRWC run are shown in Figure D-10 as plots of displacement and velocity as a function of time along with the true values and the measurements of Table D-1. This 2^{nd} order least squares regression polynomial agrees exceptionally well with the true relations for displacement and velocity having constraints imposed to satisfy equations (6) and (7).

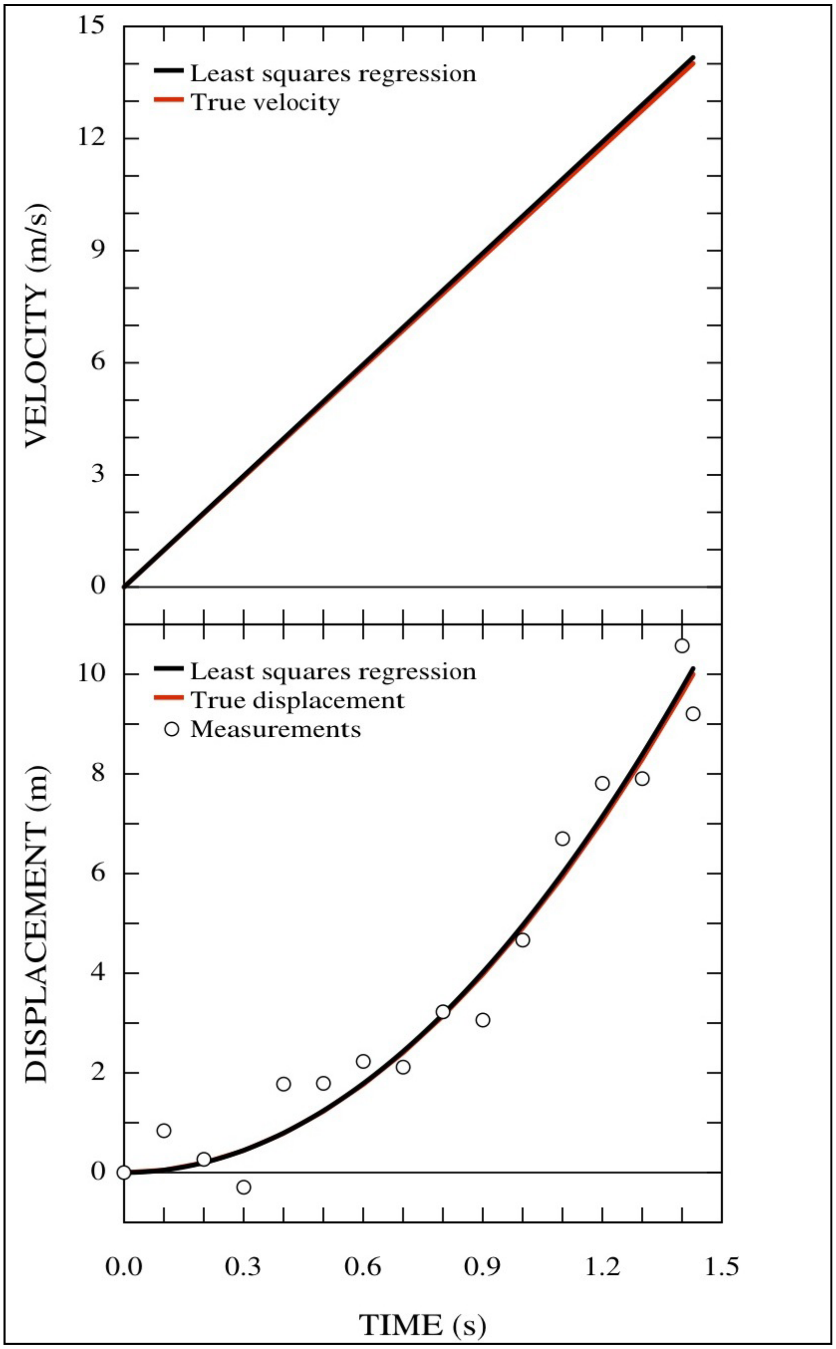


Figure D-10. Least Squares Polynomial Regression for n_p = 2 with Constraints

Indeed, the polynomial coefficients b_0 and b_1 are now zero valued, as shown in Figure D-7. Furthermore, the acceleration, $a = 2 \times b_2 = 9.92$ m/s^2, is much closer to the true value of 9.81 m/s^2 demonstrating the value of added constraints to the least squares polynomial regression procedure when they are known.

APPENDIX E
EXAMPLE 2

For this example, the LSPRWC code was applied to the representation of a set of observations artificially generated through the application of random dispersions to a hyperbolic spiral. The hyperbolic spiral in polar coordinates is given by

$$R = a/\theta \tag{1}$$

or in Cartesian coordinates

$$x = a/cos(\theta), \tag{2}$$

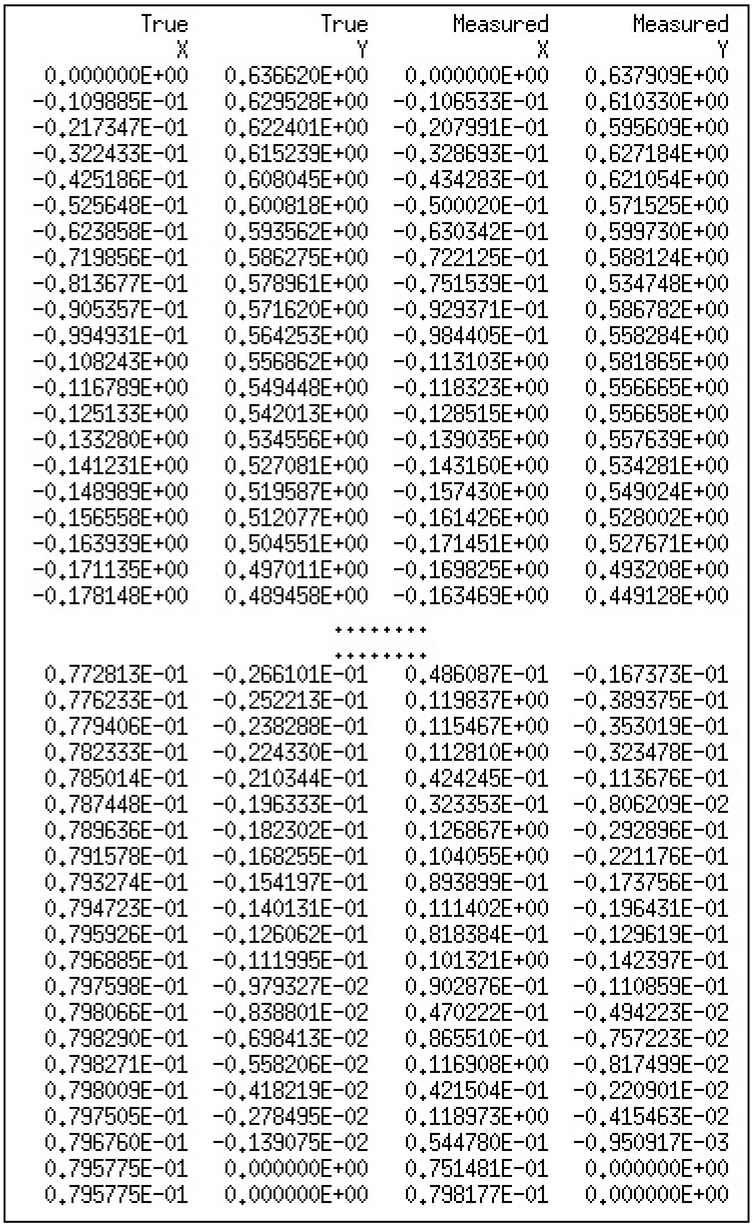
$$y = a/sin(\theta), \tag{3}$$

where

a = constant,
R = radius,
θ = angle,
x = ordinate,
y = absicca.

A partial list of the observations for this example are given in Table E-1, along with the true values for the *x* and *y* coordinates. Note that these observations were generated using equation (1) for equal 1° increments of θ from 90° through 720° with a dispersion applied to *R* within a uniformly distributed random resolution. These polar coordinates were then converted to Cartesian coordinates to produce the values given in Table E-1.

Table E-1. Observations, Cartesian Coordinates



True X	True Y	Measured X	Measured Y
0.000000E+00	0.636620E+00	0.000000E+00	0.637909E+00
-0.109885E-01	0.629528E+00	-0.106533E-01	0.610330E+00
-0.217347E-01	0.622401E+00	-0.207991E-01	0.595609E+00
-0.322433E-01	0.615239E+00	-0.328693E-01	0.627184E+00
-0.425186E-01	0.608045E+00	-0.434283E-01	0.621054E+00
-0.525648E-01	0.600818E+00	-0.500020E-01	0.571525E+00
-0.623858E-01	0.593562E+00	-0.630342E-01	0.599730E+00
-0.719856E-01	0.586275E+00	-0.722125E-01	0.588124E+00
-0.813677E-01	0.578961E+00	-0.751539E-01	0.534748E+00
-0.905357E-01	0.571620E+00	-0.929371E-01	0.586782E+00
-0.994931E-01	0.564253E+00	-0.984405E-01	0.558284E+00
-0.108243E+00	0.556862E+00	-0.113103E+00	0.581865E+00
-0.116789E+00	0.549448E+00	-0.118323E+00	0.556665E+00
-0.125133E+00	0.542013E+00	-0.128515E+00	0.556658E+00
-0.133280E+00	0.534556E+00	-0.139035E+00	0.557639E+00
-0.141231E+00	0.527081E+00	-0.143160E+00	0.534281E+00
-0.148989E+00	0.519587E+00	-0.157430E+00	0.549024E+00
-0.156558E+00	0.512077E+00	-0.161426E+00	0.528002E+00
-0.163939E+00	0.504551E+00	-0.171451E+00	0.527671E+00
-0.171135E+00	0.497011E+00	-0.169825E+00	0.493208E+00
-0.178148E+00	0.489458E+00	-0.163469E+00	0.449128E+00
	++++++++		
	++++++++		
0.772813E-01	-0.266101E-01	0.486087E-01	-0.167373E-01
0.776233E-01	-0.252213E-01	0.119837E+00	-0.389375E-01
0.779406E-01	-0.238288E-01	0.115467E+00	-0.353019E-01
0.782333E-01	-0.224330E-01	0.112810E+00	-0.323478E-01
0.785014E-01	-0.210344E-01	0.424245E-01	-0.113676E-01
0.787448E-01	-0.196333E-01	0.323353E-01	-0.806209E-02
0.789636E-01	-0.182302E-01	0.126867E+00	-0.292896E-01
0.791578E-01	-0.168255E-01	0.104055E+00	-0.221176E-01
0.793274E-01	-0.154197E-01	0.893899E-01	-0.173756E-01
0.794723E-01	-0.140131E-01	0.111402E+00	-0.196431E-01
0.795926E-01	-0.126062E-01	0.818384E-01	-0.129619E-01
0.796885E-01	-0.111995E-01	0.101321E+00	-0.142397E-01
0.797598E-01	-0.979327E-02	0.902876E-01	-0.110859E-01
0.798066E-01	-0.838801E-02	0.470222E-01	-0.494223E-02
0.798290E-01	-0.698413E-02	0.865510E-01	-0.757223E-02
0.798271E-01	-0.558206E-02	0.116908E+00	-0.817499E-02
0.798009E-01	-0.418219E-02	0.421504E-01	-0.220901E-02
0.797505E-01	-0.278495E-02	0.118973E+00	-0.415463E-02
0.796760E-01	-0.139075E-02	0.544780E-01	-0.950917E-03
0.795775E-01	0.000000E+00	0.751481E-01	0.000000E+00
0.795775E-01	0.000000E+00	0.798177E-01	0.000000E+00

The full set of observations from Table E-1 are presented in the plot of Figure E-1 and show immediately that this set of (x, y) data pairs or observations will not be suitable for a least squares polynomial regression. The true position data depicting the hyperbolic spiral, as indicated in red, exhibits locations of infinite slope which cannot be represented with polynomials. Furthermore, the (x, y) observations may well be multivalued for a given x location, as is certainly the case for the hyperbolic spiral. Although artificially generated specifically for this example, similar Cartesian coordinate data sets do occur and pose similar problems for regression.

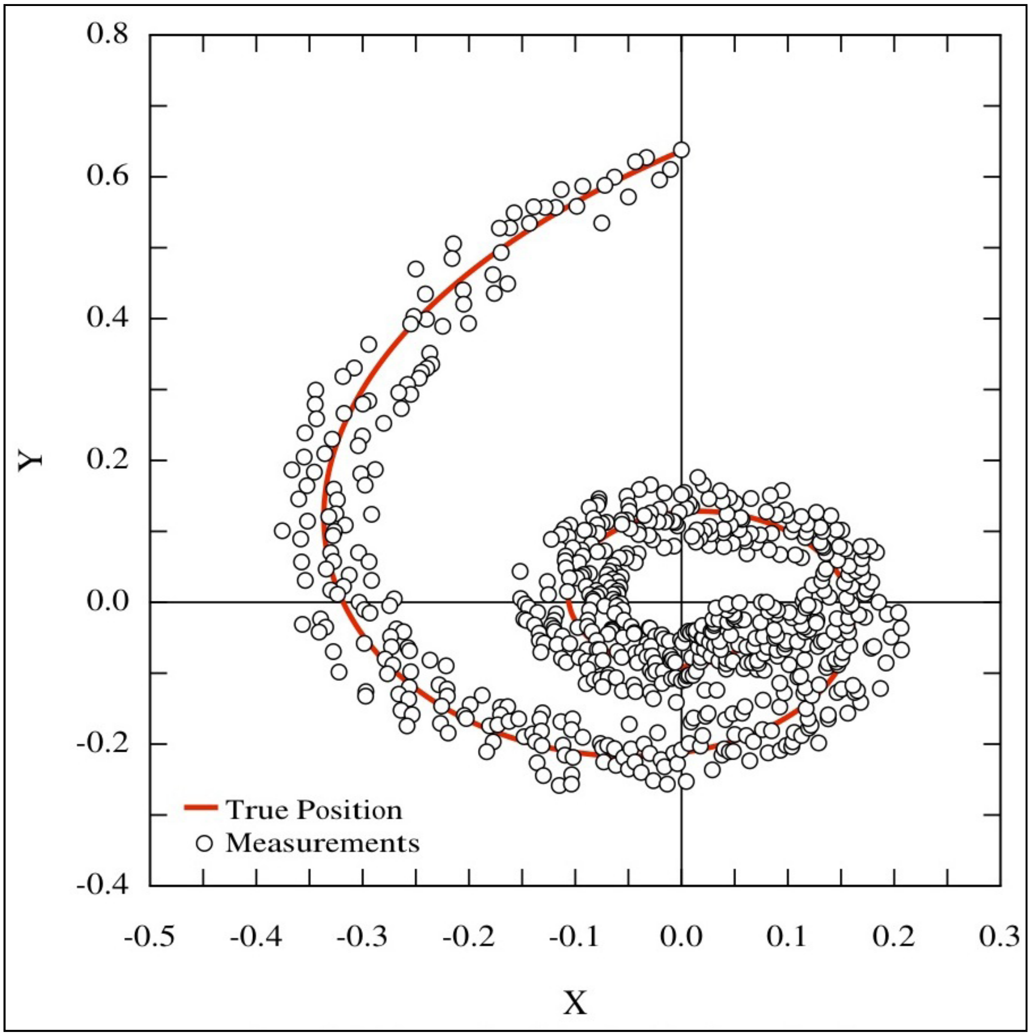


Figure E-1. Measured and True Position in (x, y) Coordinates

Something must be done to fit this (*x*, *y*) data set of Table E-1 with a least squares polynomial regression and this is accomplished, at least for this example, through a coordinate transformation. Figure E-2 shows the data in polar coordinates, data which is clearly well behaved and amenable to treatment with the method of least squares polynomial regression.

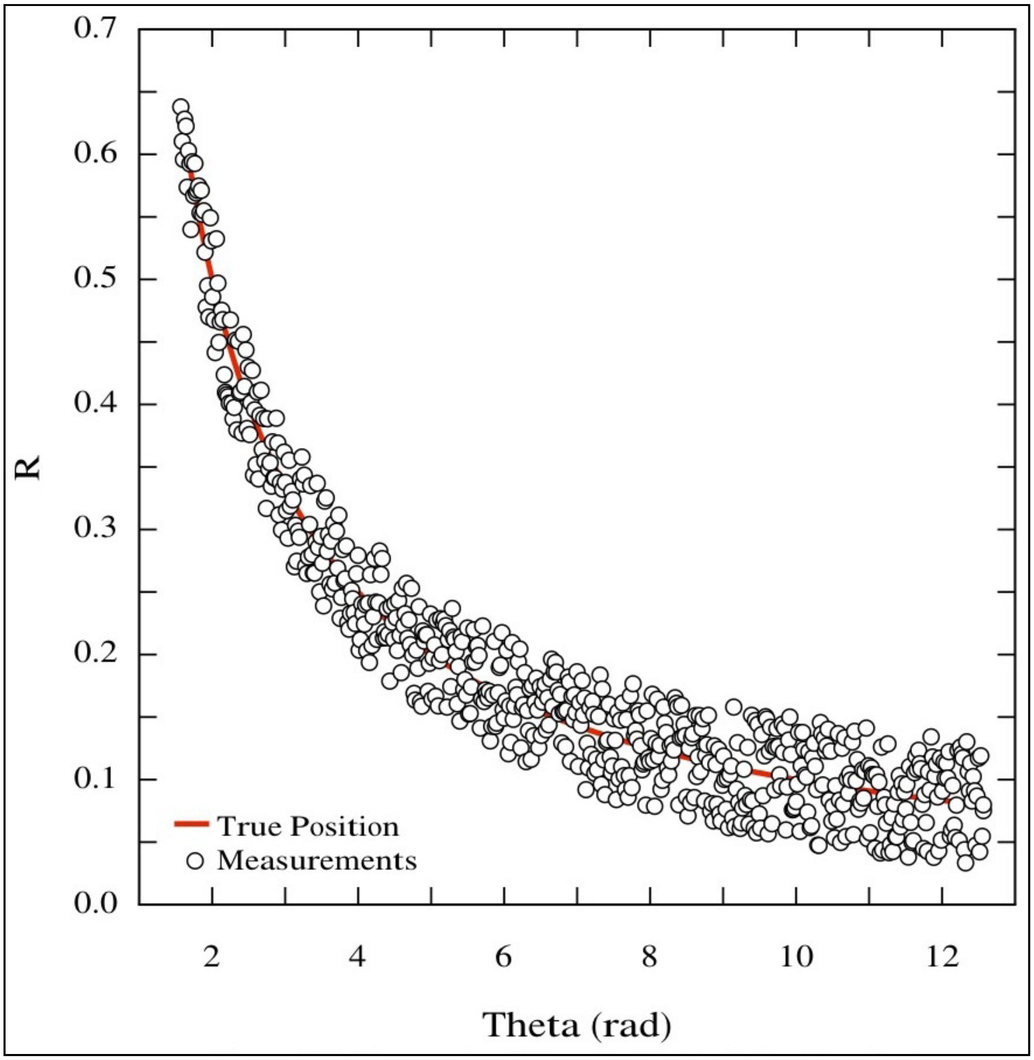


Figure E-2. Measured and True Position in Polar Coordinates

The LSPRWC code was then run with the observations in the polar coordinates of Figure E-2 for polynomial orders from 2 through 5 to produce the results of Figure E-3. This plot of *R* versus *θ* shows a reasonable fit for the regression polynomials, although none truly duplicate the hyperbolic spiral *a/θ* as shown in the figure. Perhaps this should not be unanticipated since *R* is a function of *θ* to the -1 power, while the LSPRWC code limits the n_p order regression polynomial, in this case, to

$$R(\theta) = \sum_{j=0}^{n_p} b_j \theta^j$$

for $0 \leq n_p \leq 9$.

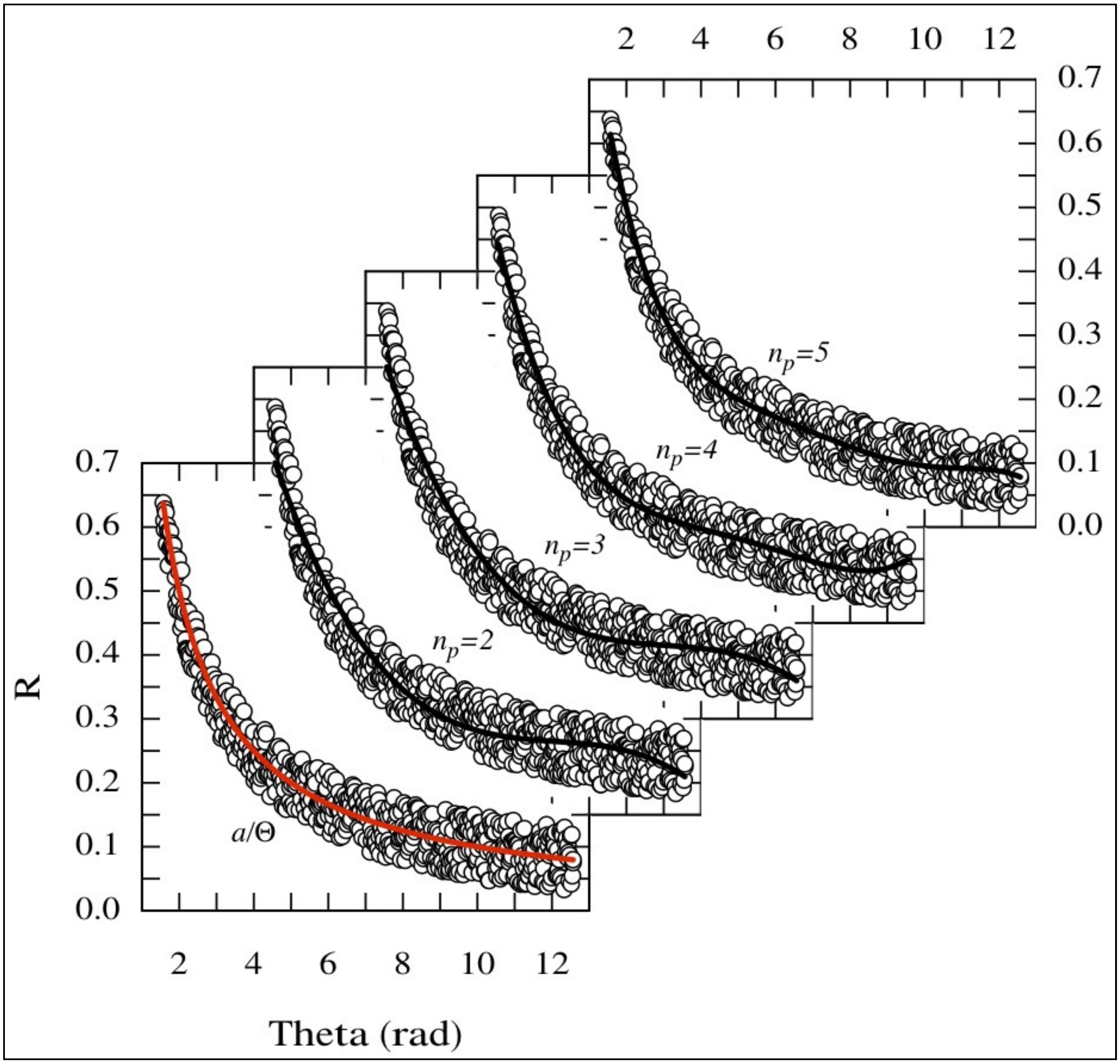


Figure E-3. Least Squares Polynomial Regression for np = 2, 3, 4, 5

Assuming the 5th order regression polynomial to give the best fit, a set of evaluations from that polynomial over the range 90° ≤ θ ≤ 720° were converted to Cartesian coordinates and plotted in Figure E-4 along with the observations and true values from Table E-1. Surprisingly, the fit is quite good with the largest departure from the theoretical occurring for values of θ near 90° rather than 720°, as might be surmised from Figure E-2.

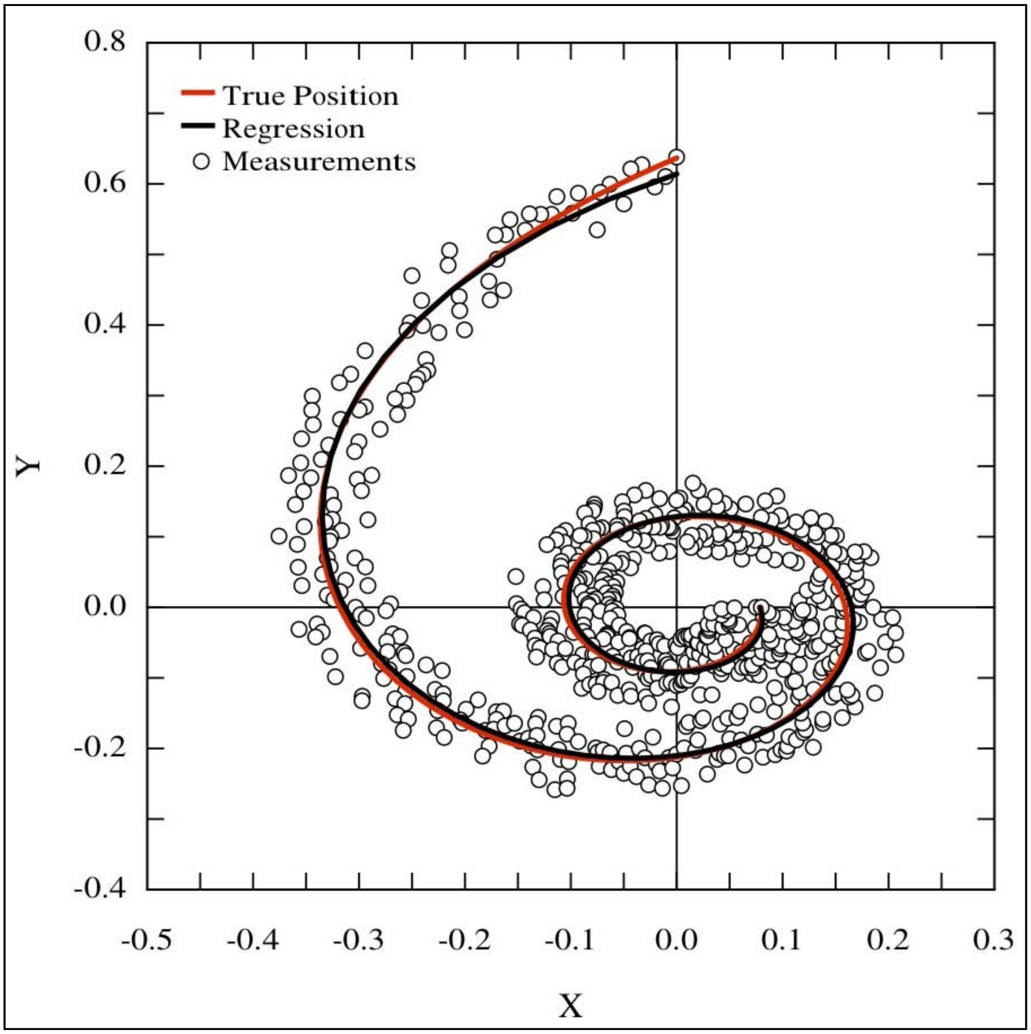


Figure E-4. Least Squares Polynomial Regression for n_p = 5

INITIAL DISTRIBUTION LIST

		Copies
Weapon Systems Technology Information Analysis Center Alion Science and Technology 201 Mill Street Rome, NY 13440	Ms. Gina Nash gnash@alionscience.com	Electronic
Defense Technical Information Center 8725 John J. Kingman Rd., Suite 0944 Fort Belvoir, VA 22060-6218	Mr. Jack L. Rike jrike@dtic.mil	Electronic
AMSAM-L	Ms. Anne C. Lanteigne anne.lanteigne@us.army.mil	Electronic
	Mr. Michael K. Gray michael.k.gray@us.army.mil	Electronic
RDMR		Electronic
RDMR-CSI		Electronic
RDMR-SS	Mr. Gregg Tackett Gregory.tackett@us.army.mil	Electronic
	Mr. Mike Eisen mike.eisen@us.army.mil	Electronic
RDMR-SSM-A	Mr. Lamar Auman lamar.auman@us.army.mil	Electronic
	Dr. Clark Mikkelsen clark.mikkelsen@us.army.mil1	Electronic/Hardcopy